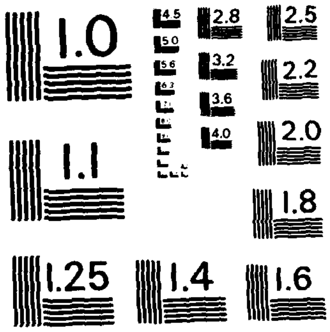
AD-A160 317 ADAPTIVE HYBRID PICTURE CODING VOLUME 2(U) ARKANSAS UNIV FAYETTEVILLE DEPT OF ELECTRICAL ENGINEERING R A JONES ET AL. 01 FEB 85 AFOSR-TR-85-0922 1/2

UNCLASSIFIED AFOSR-82-0351 F/G 9/4 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

AFOSR

0922

ADAPTIVE HYBRID PICTURE CODING

AD-A160 317

Final Report, Volume II

by

Richard A. Jones
Principal Investigator

Yogendra J. Tejwani
Senior Investigator

Department of Electrical Engineering
University of Arkansas
Fayetteville, Arkansas 72701

February 1, 1985

Prepared for the

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Under Grant No. AFOSR 82-0351

UNIVERSITY OF ARKANSAS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 85-0922
2. GOVT ACCESSION NO.: AD-A160317
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): Adaptive Hybrid Picture Coding
5. TYPE OF REPORT & PERIOD COVERED: Final Report, 1 Oct 1983 - 30 Sept 1984
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Richard A. Jones, Yogendra J. Tejwani, Carl D. Bowling
8. CONTRACT OR GRANT NUMBER(s): [redacted]1 AFOSR-82-0351
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Dept. of Electrical Engineering, 232 SE Bldg., University of Arkansas, Fayetteville, AR
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 2305/B3
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research, Bolling Air Force Base, Bldg. 410, Washington, DC 20332
12. REPORT DATE: 1 Feb 1985
13. NUMBER OF PAGES: ~~345~~ 144
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): [illegible] distribution [illegible]
17. DISTRIBUTION ST[ATEM]ENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY [NO]TES: Different aspects of this report have been presented at Globecom, Miami, FL 1983, International Conference on Acoustics, Speech & Signal Processing, 1984, IEEE Trans on Systems, Man, Cyber., IEEE Trans. on Comm. Tech.
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Data Compression, Quantizer, Shape Analysis, Feature Vector, Critical Points, Shape Space
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

A system for the machine recognition of partial shapes is described. Shape analysis methods are reviewed in context to the problem of machine recognition of partial shapes, and their limitations.

The problem of defining the critical points for shapes and partial shapes with various degrees of curvature is considered. It is shown that the critical points derived using criteria based on curvature alone are

DD FORM 1 JAN 73 1473

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT

insufficient to describe shapes represented by smooth curves. A new method of shape analysis is described which exhibits superior performance over the critical point detection methods based on curvature alone. The critical points determined by this method are based on a set of coordinate axes that are dependent on the shape itself. This guarantees that the critical points detected are independent of size, rotation, and displacement of the shape. The results of applying this new procedure to actual shapes are demonstrated and discussed.

The vector concept of shape space is introduced. This space is described in terms of it's properties. Two theorems necessary for the machine recognition of partial shapes are stated and proved using shape space properties.

The critical points are organized into structural units called feature vectors or subshape vectors using the concept of Line of Sight of a Point. The feature vectors are concatenated to form a global shape vector. Shapes are compared feature by feature using a syntactic technique which will point out if the two shapes are similar or not. Examples are given for actual shape data.

Also presented is a method for motion compensated image coding based upon a two step displacement estimation procedure. The first step utilizes a maximum a posteriori (MAP) estimator to determine the best integer displacement, while the second step requires solving for the regression coefficients that supply the same information as the non-integer portion of the displacement. This approach is a different and simplified procedure in that the integer displacement is measured first and then a linear combination of only 4 values from the previous image, shifted by the measured integer displacement, is used. This procedure differs both from the ones which measure the displacement vector D first and interpolate the previous image, and from the ones which use only linear prediction. This method is derived and results are presented for two separate forty frame digital image sequences. A sum of absolute error distortion measure is used to determine the optimal structure of the residual quantizer.

FINAL REPORT, Vol. II. Grant No. 82-0351

ADAPTIVE HYBRID PICTURE CODING

Richard A. Jones
Principal Investigator

Yogendra J. Tejwani
Senior Investigator

University of Arkansas
Department of Electrical Engineering
Fayetteville, Arkansas 72701

1 February 1985

Prepared for the Air Force Office of Scientific Research

MACHINE RECOGNITION OF PARTIAL SHAPES
using
SHAPE VECTORS

ABSTRACT

A system for the machine recognition of partial shapes is described. Shape analysis methods are reviewed in context to the problem of machine recognition of partial shapes, and their limitations.

The problem of defining the critical points for shapes and partial shapes with various degrees of curvature is considered. It is shown that the critical points derived using criteria based on curvature alone are insufficient to describe shapes represented by smooth curves. A new method of shape analysis is described which exhibits superior performance over the critical point detection methods based on curvature alone. The critical points determined by this method are based on a set of coordinate axes that are dependent on the shape itself. This gaurantees that the critical points detected are independent of size,rotation, and displacement of the shape. The results of applying this new procedure to actual shapes are demonstrated and discussed.

The vector concept of shape space is introduced. This space is described in terms of it's properties. Two

theorems necessary for the machine recognition of partial shapes are stated and proved using shape space properties.

The critical points are organized into structural units called feature vectors or subshape vectors using the concept of Line of Sight of a Point. The feature vectors are concatenated to form a globlal shape vector. Shapes are compared feature by feature using a syntactic technique which will point out if the two shapes are similar or not. Examples are given for actual shape data.

TABLE OF CONTENTS

LIST OF FIGURES

MACHINE RECOGNITION OF PARTIAL SHAPES
USING
SHAPE VECTORS

CHAPTER I.

INTRODUCTION

THE PROBLEM AND THE SYSTEM MODEL

There are many practical applications where it is necessary to identify objects or shapes in successive scenes. One such application is in the area of robot vision. Here, an object may have to be tracked from the instant the object enters the field of view, of the robot, to the time the object leaves it's field of view. Another application is in the data compression area of communication systems, where the picture data has to be transmitted over a channel to a receiver. The picture data usually consists of sets of sequences of images of scenes, where some kind of motion (activity) has taken place. The sequence of images in every set contains some 'common information', which enables the ultimate user (human viewer) to identify the set as belonging to the same scene. Data can be compressed for the purpose

of efficient transmission, if this 'common information' can be identified and transmitted along with the first frame in the sequence, while subsequently, instead of transmitting complete frames, only the 'uncommon information or changes in the scene are transmitted. A transmitter which is capable of achieveing this goal will require a 'smart receiver' which would be capable of combining the common and uncommon information and be able to simulate, predict or reconstruct the pictures of the scene to the desired human fidelity criterion.

The above process of extracting information can be divided into three steps or subprocesses, which are,

1) The correspondence process (the extraction of the common information) [88],

2) The interpretation process (the extraction of the uncommon information) [88], &

3) The predicition and simulation process.

For a communication system the first two are exclusively the domain of the 'smart transmitter', while the last is that of the 'smart receiver'. In a robot vision system, the functions of the 'smart transmitter' and the 'smart receiver' may be accomplished at the same place.

The correspondence process or problem is that of identifying a portion of changing visual array as representing the same scene or object in change or motion.

The extraction of the uncommon information about the sequence, or the interpretation process can be:

1) a qualitative process or

2) a quantitative process

The process is a qualitative process when the extraction of information is based on statistics derived by operations on adaptive blocks or blocks of variable size, within the picture and the final output results in statements like, ' the object located near the left conner of the picture went through a rotation of 1.57 radians about the z axis while the background moved forward by about a feet'.

The process is a quantitative process if the extraction of information is based on statistics, derived by operations on predefined or fixed blocks or sub-blocks of the picture. e.g. operations based on rows or columns . The representation of such information is not qualitative. Usually such a representation is in the form of numbers of data arrays which cannot be directly interpreted. e.g. the 2-D array representating

the difference in the pixel intensities between two frames in the sequence.

The correspondence process begins before the object has completely entered the field of view and continues even while the object is undergoing occlusion or leaving the field of view. This suggests that a correspondence process based on global, shape or object recognition methods alone would not be adequate for such a system. In this dissertation, a system which is capable of identifying an object, when it is partly or partially in field of view, with itself at some other time, when it is completely in the field of view, will be called a Partial Shape Recogntion System. In such a system recognition of shapes is based upon identifying elements or correspondence tokens in different views as representing the same shape in at different times, and thereby maintaining the perceptual identity of objects in motion.

The correspondence tokens which are matched by the correspondence process [88] include critical points, edge fragments, bars, small blobs etc. These tokens are detected first and organised hierachally into more structured forms, and finally into distinct objects.

The block diagram of a Partial Shape Recognition System is shown in Fig. 1. The input to the system is a digitized image of the scene. Refering to Fig. 1. the output of the image processor can be divided into two groups 1) data represented by lists, trees or directed graph structures e.g. sequence of boundary points describing a shape. 2) data represented by block like structures e.g. average color or intensity information of regions.

In this dissertation it will be assumed that the output of the Image Processor is availabe. Furthermore, attention will be restricted to the data of the first type, or the area in the block diagram in Fig. 1. enclosed by the dotted line. Specifically it will be assumed that the sequence of boundary points describing the shape is availabe.

As a first step the system of Fig. 1. linearly interpolates a curve between the sequence of points describing the boundary of the shape. This curve is then resampled uniformly at constant arc lengths. The degree of the interpolating curve may subsequently be altered depending on knowledge about the shape obtained from the data base, or with the help of information through the feedback path shown in Fig. 1. The

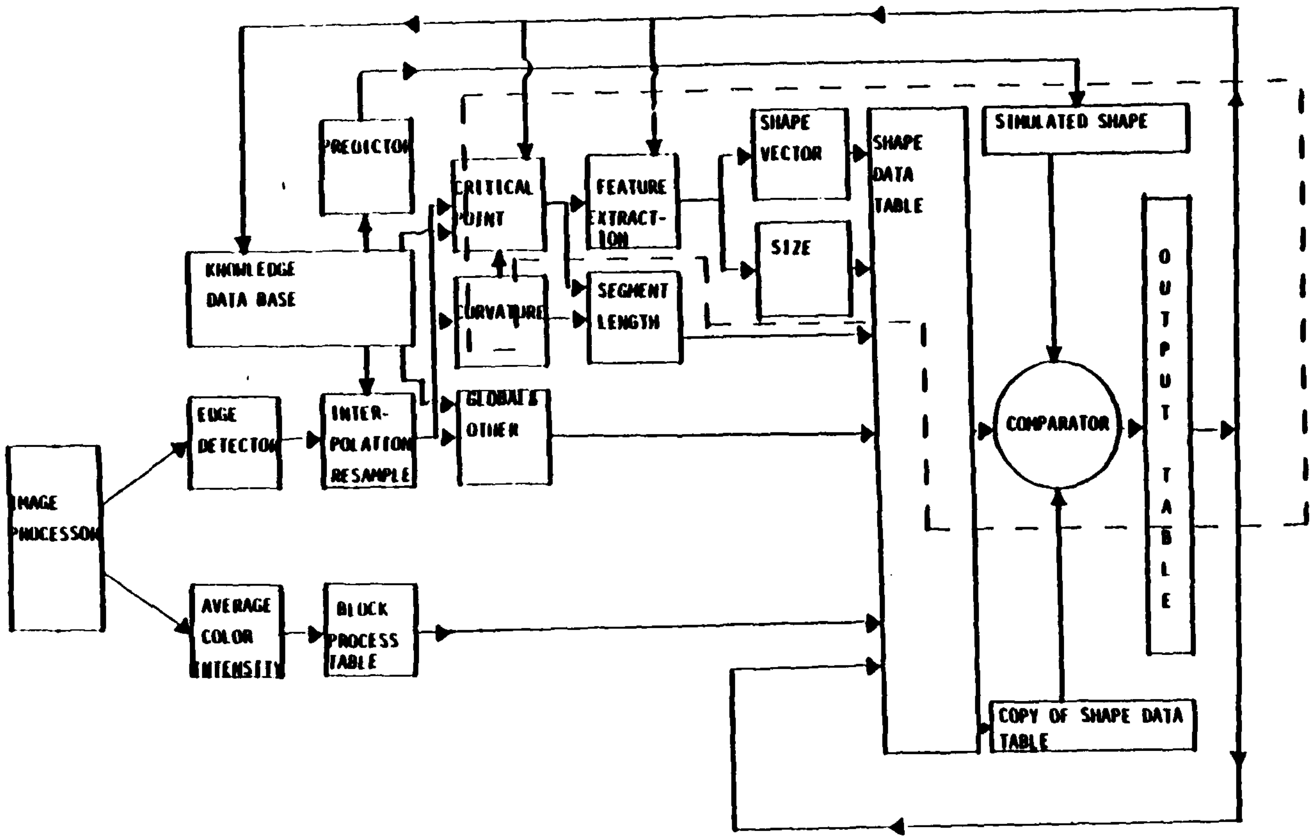


FIG. 1. BLOCK DIAGRAM OF PARTIAL SHAPE RECOGNITION SYSTEM.

information necessary to alter the degree of the interpolating curve can only be obtained after the shape data has been processed at least once. The interpolated curve is then examined for evaluation of critical points and curvature. The combination of these two factors helps in the extraction of features and the length of segments to be matched. These features may be altered deleted or merged based on the knowledge from the data base or the feedback from the output table. Though often, features may be deleted based on their relative size, the rules for altering the feature structures are not well established and will not be addressed.

The features are then converted into shape vectors and stored in the data table. Information about the segments of required length is also stored in the shape data table. This information can be in the form of curvature of the segments, length of segments, interconnections between segments, frequency domain information about curvature, and or, other space invariant properties.

The shape data table which also includes information about block processes is then compared with the shape data table simulated by the predictor with knowledge from the data base, and also with a copy of itself.

Comparison with the latter helps in extraction of symmetry and other aspects of the shape. The results of comparsion are stored in the output table. These results are fed back into the various blocks of the system shown in Fig. 1. The feedback is mainly to establish relationships with other shapes on the scene, to delete relatively less important critical points or features, to update the data base, and to the change the degree of the interpolating curve if necessary.

At the present the methods of recognizing shapes [20], [21], [60], [65], [63], [66], [91], can be categorized as either global or local in nature. Within the class of global shape analysis methods, there are two categories that under certain circumstances possess the ability to recognize complete or whole shapes independent of size, rotation, or location. These are the Fourier descriptors methods and the Syntactic or the Graphical methods. The Fourier descriptors based algorithm performs satisfactorily on complete shapes. The Fourier coefficients extracted, are indeed independent of size, rotaion, and location when the shapes are complete. However this method does not perform satisfactorily and in fact fails entirely when the class of shapes is allowed to include incomplete or partial shapes. An example is presented in chapter II

that demonstrates that Fourier descriptors method fails to work on incomplete shapes. The results of the experiment are discussed in order to point out specifically why the algorithm cannot perform satisfactorily on partial shapes.

The other class of global methods namely the Syntactic methods [21], have restricted use in recognition of shapes because these algorithms tactily assume a priori that the shapes have been identified by their parts. These algorithms then investigate the relationship between the various parts of the shape. A human shape, for instance has a hand or a face at some definite orientation and location with respect to each other.

The Local category of shape analysis algorithm [91], uses curvature as a criterion for detecting the peaks and valleys of a shape. These peaks and valleys are called the local shape descriptors. This shape comparision algorithm is not independent of rotation. In Chapter II specific examples are given that demonstrates that the present algorithm is not independent of rotation.

In Chapter III. the concept of curvature is presented form the point of view, of differential geometry. The

concept is presented so as to determine, why the local shape descriptors in Chapter II are not independent of rotation. Further it is shown that the critical points found by using curvature as a criterion are insufficient to describe many shapes. This is demonstrate by an example.

In Chapter IV the concept of Line of Sight of a Point (LSP) and Line of Sight of a straight line Axis (LSA) are introduced. These concepts are then used to define to Adaptive Line of Sight method (ALS) for determining the critical points. The effect of this algorithm on actual shape data is presented. The critical point detected by this method are very close to those perceived by the human vision system in most cases. The ALS method can also be used as a basis for detecting the axes of symmetry in a shape, if any exist. However, such a task can be only be achieved at a post cognitive level; i.e. the critical points found by the ALS method can be used to locate the axes of symmetry. Location of such an axes is an important part of shape analysis, because, it seems that in the human vision system ,the critical points located with respect to an axis of symmetry are considered more important than critical points located with respect to any other axis. Unfortunately the ALS method cannot always detect an

axis of symmetry at the precognitive level.

In Chapter V concepts from several areas of shape analysis [19], [52], [91], are combined with some entirely new concepts concerning shapes for the purpose of providing the foundation for a new approach to define shape as vectors in an appropriate space. The properties of this space are stated definitively, after the concept of size variable has been solidified. This vector space is called the shape space. Two theorems usefuls in the partial shape recognition problem are stated and proved utilizing shape space properties.

In Chapter VI the concept of Line of Sight of a point is used to organize critical points into feature vectors in the shape space. These feature vectors are then concatenated to form shape vectors. Procedures and tests necessary for comparing shape vectors are examined. The comparison procedure is based on a Syntactic method which will point out whether one shape is part of a more complex shape, or whether the shapes are totally dissimilar.

The conclusion and discussion in Chapter VII places into perspective the overall effectiveness of methods for analyzing shape based on critical points, addresses the question of thresholds, the problem of locating axes

of symmetry in shapes, and the need for relocation of critical points based on such axes.

CHAPTER II.

OTHER METHODS AND THERE RELEVANCE TO THE PROBLEM

II.a FOURIER DESCRIPTORS METHOD

There exist two types of Fourier descriptors. The first type of descriptors, used by Zahn and Roskies [98], have been called descriptors S_n by Pavlidis [60]. In the method of descriptors S_n the shape is represented by the continuous function,

$$a(\, t(k), k) = \phi(\, k\,) + t(\, k\,) \qquad \text{(II.1)}$$

where $t(\, k\,) = 2\pi \ell(k)/L$

$\ell(\, k\,) =$ arc length between the starting point and the k th point on the curve.

$\phi(\, k\,) =$ net amount of angular change between the starting point and the k th point on the curve.

$L =$ the perimeter of the curve

The descriptors S_n for a continuous shape are then defined as,

$$S = \frac{1}{2\pi}\int_0^{2\pi} a(t(k),\, k)\exp(-j2\pi nt)\,dt \qquad \text{(II.2)}$$

These descriptors exhibit some notable shortcomings. Among these are 1) the property of the closure of the curve is not preserved 2) simple shapes such as squares and triangles cannot be distinguished from one another when only the vertices are given. The second type of Fourier descriptors, namely the descriptors T_n [60], [63], [66] exhibit charactersitics that are superior to the descriptors S_n in the sense that the reconstruction of the shape from a finite set of coefficients leads to a closed curve. Also the convergence properties are superior.

For the descriptors T_n the shape data is represented in the complex form,

$$u(\ell) = x(\ell) + j\, y(\ell) \qquad \text{(II.3)}$$

where $(x(\ell), y(\ell))$ are the coordinates of the point on the curve and ℓ is the arc length from the defined starting point. The descriptors T_n are then given by,

$$T_n = \frac{1}{L}\int_0^L u(\ell)\exp(-j2\pi n\ell/L)\,dl \qquad \text{(II.4)}$$

The Fourier descriptors based algorithm normalizes for position by setting T_0 to zero . Normalization for scale, rotation, and starting point of a contour is achieved by multiplying the n th coefficient by

$s \exp(j(\phi + n\alpha))$. The parameter s scales the shape to the normalized size, the parameter $(\phi + n\alpha)$ rotates the contour to the normalized position. The normalization is such that the coefficients T_{+1} and T_{-1} are pure imaginary numbers and their sum has magnitude one.

It is easy to see why such a normalization is necessary for discrete data. In this case, the Fourier descriptors are given by,

$$T(n) = \frac{1}{N}\sum_{k=0}^{N-1}(x(k) + j\,y(k))\exp(-j2\pi nk/N) \tag{II.5}$$

Multiplying (I.b.5) by $s \exp{-j(\phi + n\alpha)}$ the normalized coefficients, $T_N(n)$ are then given as,

$$T_N(n) = T(n)\, s \exp j(\phi + n\alpha) \tag{II.6}$$

Next imposing the requirement on (I.b.6), that the coefficient be purely imaginary at $n = \pm 1$ leads to the condition that,

$$\frac{2\pi k}{N} - (\phi + \alpha) = 2m + \left(-\frac{2\pi k}{N} - (\phi - \alpha)\right) \tag{II.7}$$

where m is an integer.

Setting the real part to zero by appropriate choices of ϕ and α is equivalent to normalizing for rotation and starting point of the contour. As a matter of fact, the two equations could have been set equal to any constant.

The imaginary parts of the normalized coefficient at $n = \pm 1$ are,

$$IM(T_N(1)) = \frac{s}{N}\sum_{k=0}^{N-1}\left(-x(k)\sin\left(\frac{2\pi k}{N} - (\phi+\alpha)\right) + y(k)\cos\left(\frac{2\pi k}{N} - (\phi+\alpha)\right)\right) \quad \text{(II.8)}$$

$$IM(T_N(-1)) = \frac{s}{N}\sum_{k=0}^{N-1}\left(-x(k)\sin\left(\frac{2\pi k}{N} - (\phi-\alpha)\right) + y(k)\cos\left(\frac{2\pi k}{N} - (\phi-\alpha)\right)\right) \quad \text{(II.9)}$$

Summing and equating the magnitude, of the normalized coefficients at ± 1, to one, yields,

$$s = \left(\sum_{k=0}^{N-1}\left(x(k)\sin(A+B) - \sin(A-C)\right) + \left(y(k)\left(\cos(A-C) + \cos(A-B)\right)\right)\right)^{-1} N \quad \text{(II.10)}$$

where $A = 2\pi k/N$

$B = \phi + \alpha$

$C = \phi - \alpha$

Another function [64] which has also been used for normalization is the standard deviation of the data,

$$\sigma = \left[\frac{N}{\sum_{k=1}^{N}(x(k)-\bar{x})^2 + (y(k)-\bar{y})^2}\right]^{-\frac{1}{2}} \quad \text{(II.11)}$$

where $\bar{x} = \frac{1}{N}\sum_{k=0}^{N} x(k)$

$$\bar{y} = \frac{1}{N}\sum_{k=0}^{N} y(k)$$

It is apparent that both s^{-1} and σ are homogeneous function of order one of the data points.

In a real shape recognition application it is not known a priori whether the shape under examination is a part of a more complex shape or a shape in its own right. The above descriptors in their present form are not capable of recognizing that a partial shape may be part of a more complex shape. An example is presented to demonstrate the validity of this contention. The shapes used are shown in Fig. 2. Figure 2-a is a complete shape, namely a swept wing plane, while Figure 2-b is a partial shape, namely the front part of the plane. The plots of the real and imaginary part of the normalized Fourier descriptors obtained by using (II.10) for the complete shape are shown in Fig. 3-a. The corresponding plots for the partial shape are shown in Fig. 3-b. Similar data obtained using (II.11) is shown in Fig. 4. It is apparent that comparing the two sets of data yields nothing more that a statement that the two shapes are dissimilar. Two explanations for this are 1) the

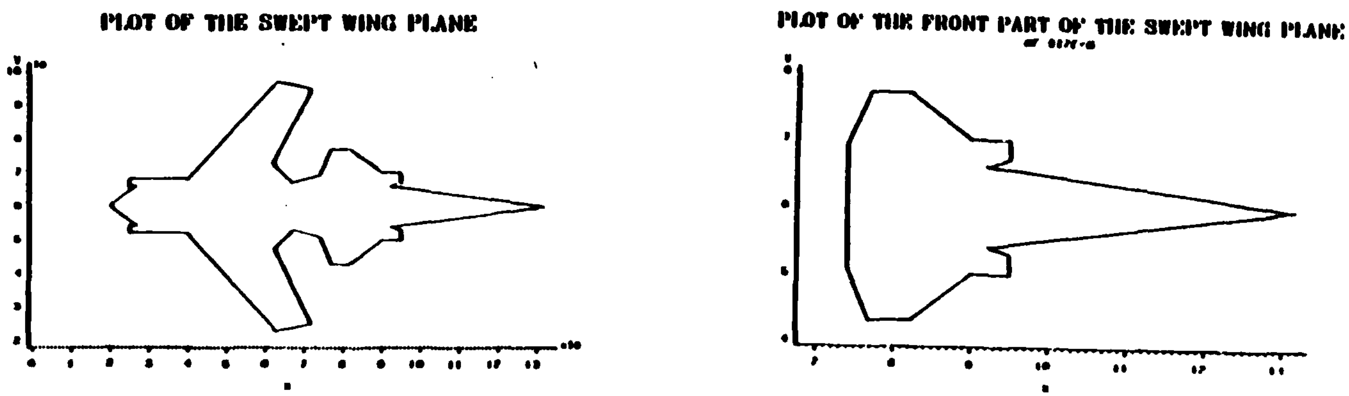


FIG. 2. SWEPTWING PLANE SHAPES.

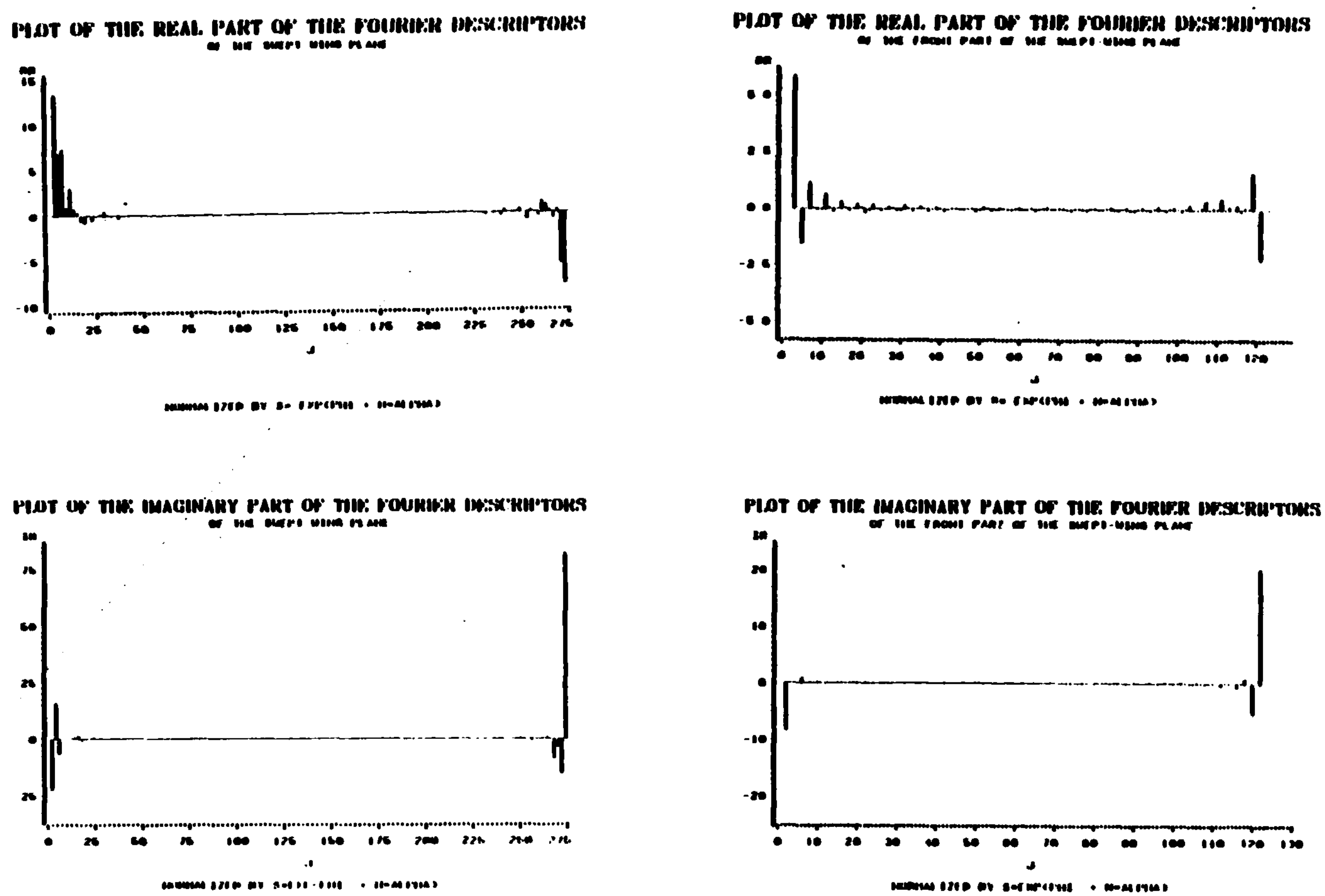


FIG. 3. FOURIER DESCRIPTORS CALCULATED USING THE PARAMETER s.

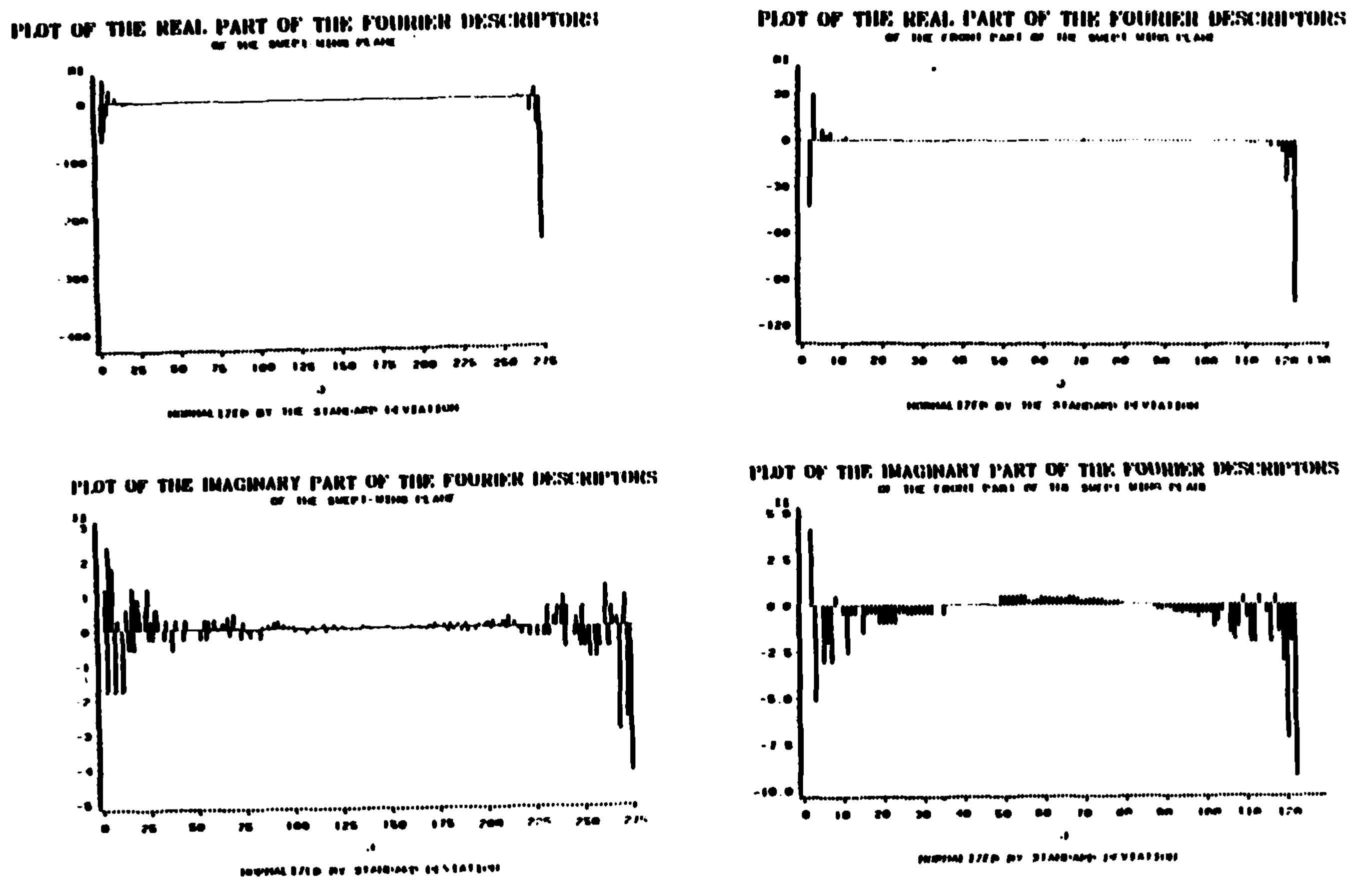


FIG. 4. FOURIER DESCRIPTORS CALCULATED USING THE PARAMETER σ.

parameters $\bar{s}^{1}$ and σ are not independent of the shape and 2) the Fourier descriptors method compares the 'frequencies' of the shapes. The frequencies of the partial shape are not the same as the frequencies of the complete shape.

Besides the above disadvantages there is a very important aspect of the Fourier descriptors that seems to have skipped the attention of most researchers in the field. The fundamental reason for comparing shapes using Fourier descriptors lies in the fact that the Fourier coefficients for a given shape are unique; however there are many transforms that will give a unique set of coefficient for a shape. Considering the computational power of the present day computers there is no reason to treat other transforms as second class citizens.

II.b LOCAL SHAPE DESCRIPTORS

Curvature has been used to determine representative points on a shape by many researchers [14], [91]. Usually curvature is defined as an operation on three points in a sequentially ordered list.

The local category of shape analysis algorithm [91] uses 'curvature' as a criterion for detecting the peaks and valleys of a shape. These peaks and valleys are called the local shape descriptors. This shape comparision algorithm is not independent of rotation. A specific example is given that demonstrates that the present comparision type local shape analysis algorithm is not independent of rotation. The definition of curvature as given in [91] is examined and used to find peaks and valleys (critical points) of a simple shape (cardioid), before and after rotation.

The Local Shape descriptors method has been used [91], for comparing shapes stored in a library. In this method, the angle function is defined as,

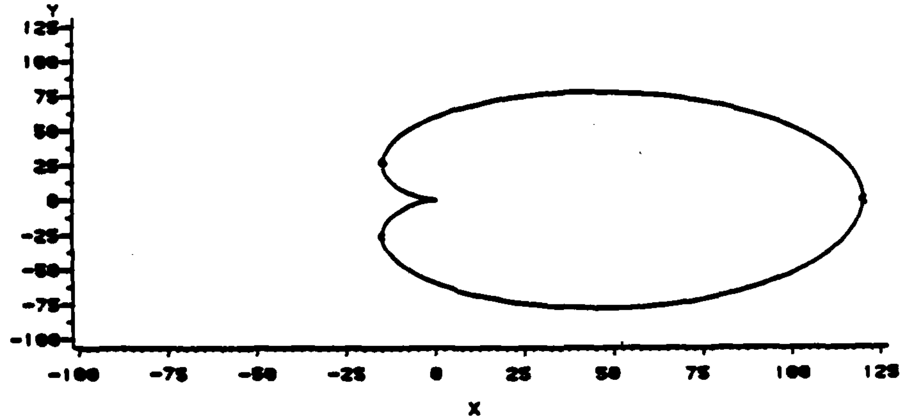
$$s(k)=\arctan((y(k)-y(k-1))/(x(k)-x(k-1))) \qquad (II.12)$$

Curvature is then defined as the derivative of the angle function, that is $s'(k)$. The peaks and valleys in $s'(k)$ are used for finding the peaks and valleys of the

curve. Each local shape descriptor then consists of two adjacent peaks and a valley (alternate angles and a distance).

It is apparent from (II.12) that this procedure results in a nonlinear transformation on the data, also, the function is not finite at ± 1.57 radians. It may be generally stated that this function may be used to locate peaks and valleys where : 1) the angle between any three adjacent points is acute and 2) the shape becomes parallel to the y-axis. These are equivalent to the points where the angle becomes ± 1.57 radians with respect to the x-axis. For example consider, the cardioid shown in Fig. 5-a, this shape was generated by sampling at constant intervals of (6.28/256) radians rather than the arc length. This prevents an acute angle from occuring between any three adjacent points on the shape. The cardioid was then rotated by 1.57 radians. The corresponding plots for the angle, the derivative of the angle function, and the peaks obtained using this method are shown in Figures 5-a, 5-b, 6-a, and 6-b, respectively. Note that the peaks obtained by this method are not unique and are dependent on the rotation. Every different rotation will yield a different set of peaks. Spurious peaks will occur at the discontinuities of the arctan function that may be ± 1.57 radians or

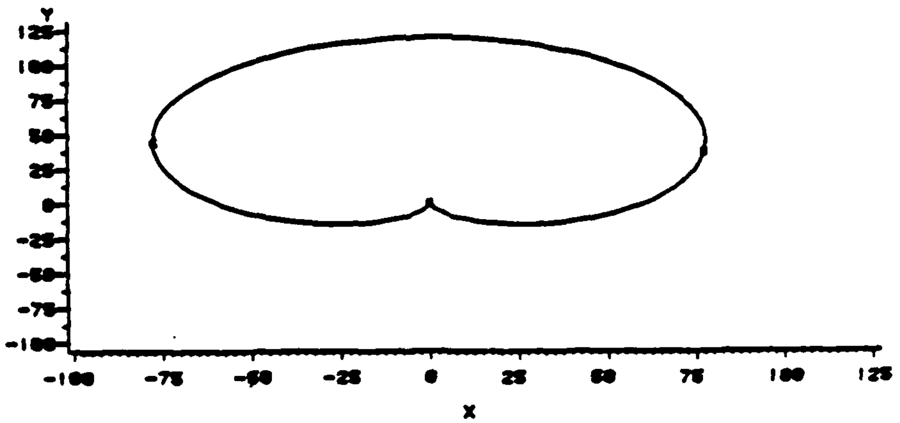
CARDIOID OF SIZE=A



5-a

C=CRITICAL POINTS FOUND BY TAKING THE DERIVATIVE OF THE ANGLE

CARDIOID OF SIZE=A ROTATED BY 1.57 RADIANS



5-b

C=CRITICAL POINTS FOUND BY TAKING THE DERIVATIVE OF THE ANGLE

FIG. 5. CRITICAL POINTS OF THE CARDIOID SHAPE.

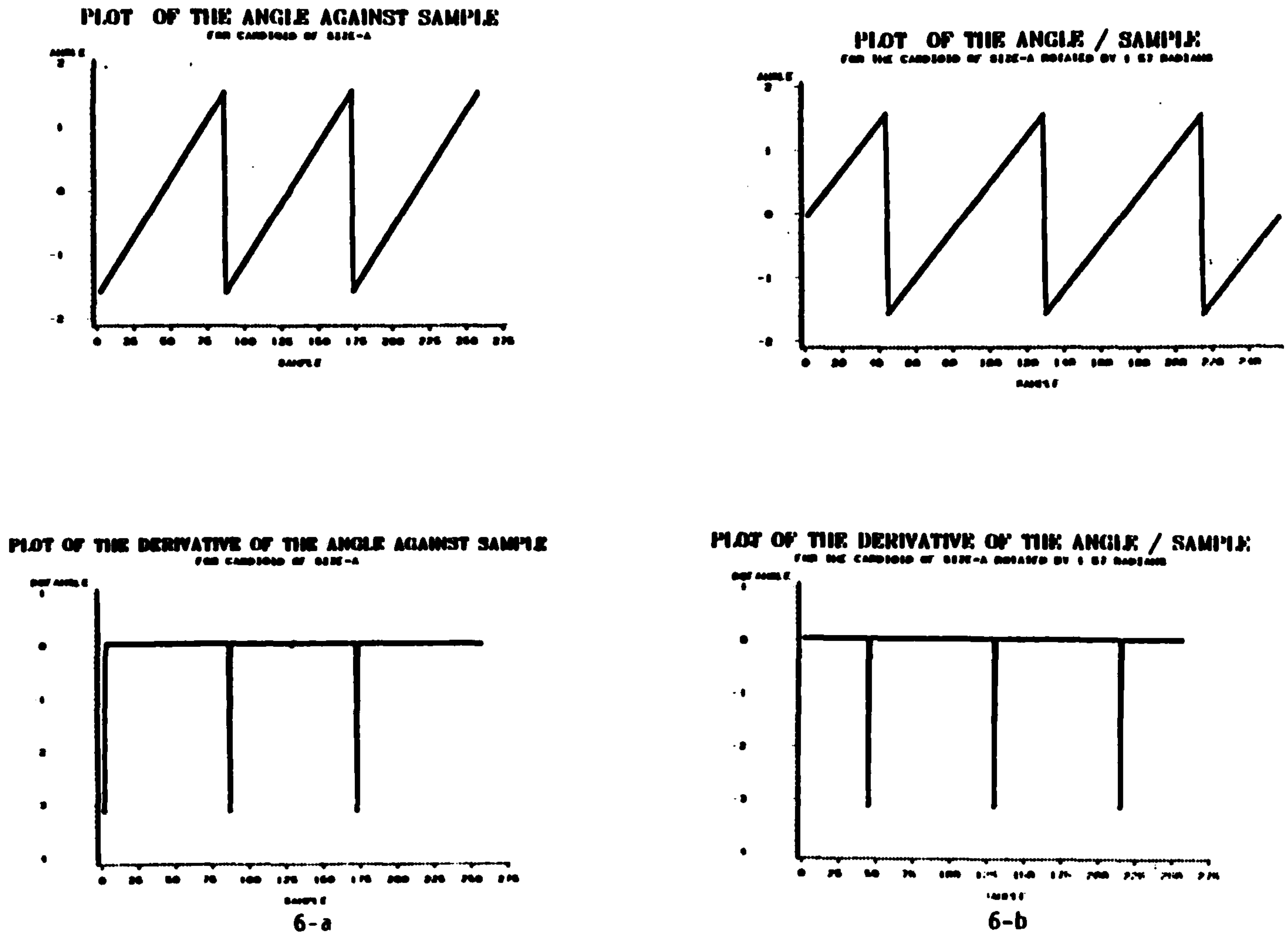


FIG. 6. ANGLES AND DERIVATIVES OF ANGLES FOR THE CARDIOID SHAPE.

II.c MISCELLANEOUS OTHER METHODS

Several methods have been utilized in the past for determining critical points [14],[19],[32]. Most of these methods are based on operations on a fixed number, of points. These methods were originally meant for ideal curves (noise free curves), where the operations were defined on a set of adjacent points. To by pass the effect of noise, and to arrive at a better estimate of the parameters being evalued, these operations were redefined, [14], [19], for points of the shape located at some fixed distance (instead of adjacent points). Since it is unlikely that an intelligent machine will have a priori knowledge about the size of the shape to be analyzed or the relationship of the number of sample points to the size, the ambuiguities involved in detection of critical points using methods that are totally dependent on operations on fixed number of adjacent points, are high. A brief description of some of these methods which are represented here by operations on three adjacent points is presented next.

II.c.i THE METHOD OF CENTROIDAL VECTORS

In this method [19], the ith point on a shape at a vector distance $d_i \angle \alpha_i$ from a reference point (typically the shape centroid) is said to be critical with respect to the reference point if

$$((d_{i-1} - d_i) \text{ and } (d_i - d_{i+1}))$$

or

$$((\alpha_{i-1} - \alpha_i) \text{ and } (\alpha_i - \alpha_{i+1}))$$

have opposite sign.

This operation is very local in nature and is extremely sensitive to noise. Round-off or truncation errors also have a deleterious effect on the operation. It also fails very often when dealing with smooth curves or in situations where the centroid is located away from the shape boundary as shown in Fig. 7.

II.c.ii METHOD OF CURVATURE VECTORS.

This method is basically a two pass process. In the first pass, the i th point on the shape with a curvature

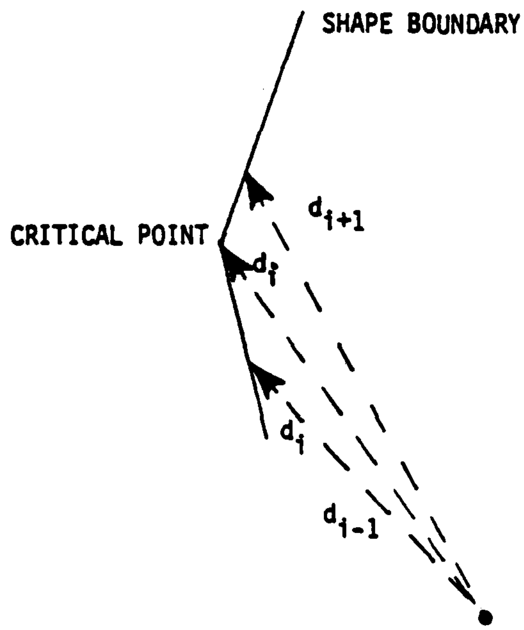


FIG. 7. A CASE AGAINST THE CENTROIDAL METHOD.

is set as a critical point if

$$(\kappa_i - \kappa_{i-1}) \text{ or } (\kappa_i - \kappa_{i+1})$$

have opposite sign. In the second pass, the points at a maximum distance from the straight line joining every two successive critical points is set as critical. This method like the previous method, is very sensitive to noise. Though it works for most of the smooth curves, there are instances where it fails. For example, this method produces only one critical point for the cardioid shape of Fig. 5.

II.c.iii THE MAGNITUDE OF CURVATURE METHOD

In the Magnitude of Curvature Method the ith point is determined to be critical if,

$$|\kappa_i| = \text{Threshold}$$

The critical points were determined by using this method on the elephant shape, which was obtained by non-uniformly sampling a hand sketched figure. These critical points are shown in Fig. 8. It is apparent that some critical points were missed. For example the critical points for the trunk are missing. This procedure is more immune to the noise than the others;

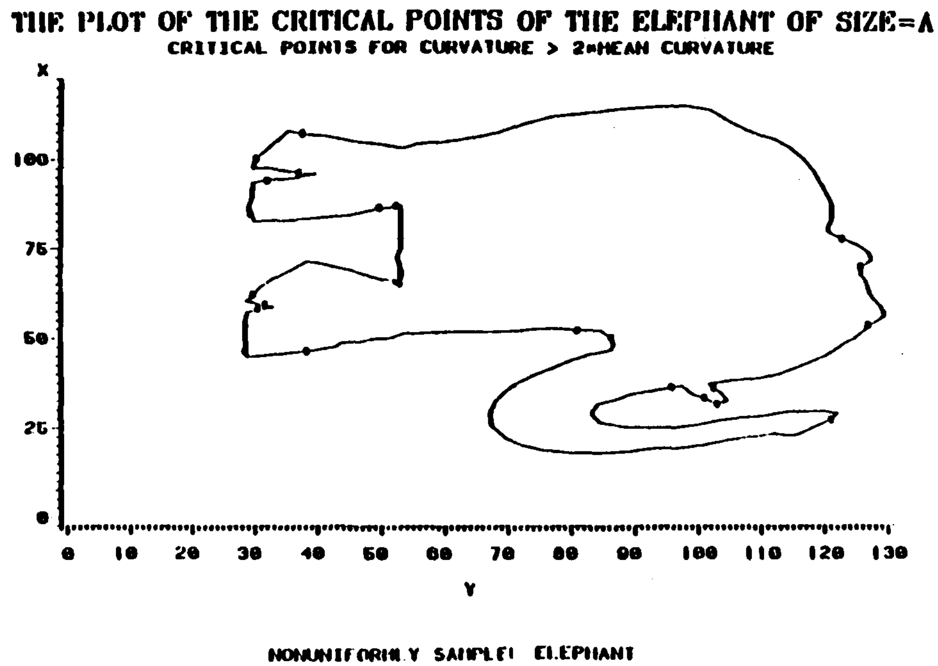


FIG. 8. CRITICAL POINTS FOR THE ELEPHANT-SHAPE.

however, an equally critical problem is added. Specifically, a threshold must be determined a priori. Additionally the method fails completely on smoothly varying curves such as the cardioid of Fig. 5.

II.c.iv THE LINE OF SIGHT METHOD.

A procedure that performs well for polygonal shapes and curves with very few boundary points is called the Line of Sight method. This method has not appeared before in literature and is being presented for the first time. It is described here to give a feeling for the problem of locating critical points on a shape. It was also the method which led to the developement of the concepts of Line of Sight and the Adaptive Line of Sight method described in subsequent chapters.

In this method, if d_i denotes the normal distance of the i th point from a straight line L, then the i th point on the shape boundary is said to be critical with respect to the straight line L if

$$(d_{i-1} - d_i) \text{ and } (d_i - d_{i+1})$$

have opposite sign. The set of critical points found with respect to the set of tangent lines drawn at all

points on the shape, will then be called the critical points. It is obvious that for shapes other than polygonal an infinite number of axes or tangent lines are required. Nonetheless, the attributes exhibited by this method are very desirable. It is however necessary to reduce the dimensionality of the problem; this is the subject that is addressed by the Adaptive Line of Sight method.

CHAPTER III.

III.a THE NEED FOR DETECTING CRITICAL POINTS.

The classical method of comparing curves or shapes in space using the geometric invariant parameters, curvature and torsion is examined here. This is done to demonstrate a need for detecting critical points based on criteria other than curvature, and to pin-point the difference between the definition of curvature as used in the field of differential geometry and the definition resulting from (II.12) The weakness of using curvature as a criterion [91], for comparing practical shapes is briefly mentioned. The cardioid shape is used again, to show that even with the differential geometric definition of curvature, curvature cannot be used as a single entity for detecting critical points which may be used for comparison of shapes.

III.b CURVES IN THREE DIMENSIONAL SPACE

The problem of shape recognition is analogous to recognition of curves in space. Therefore, well known concepts and theorems from differential geometry can be

utilized in shape analysis. Let c be a curve in Euclidian space R^n of class $C >= 1$ whose domain is I_s, where s is the parameter. Then a differential geometry theorem that is particularly useful is [40],

Theorem: Every regular curve $c: I_s \rightarrow R^n$ can be parameterized by its arc length.

In other words, given a regular curve $c: I_s \rightarrow R^n$ there is a change of variables $\theta : I_s \rightarrow I_\theta$ such that $|(c.\theta)'(s)| = 1$, where $(c.\theta)$ is a composite function.

Let $c = c(s)$ be the parametric representation of the curve under analysis with s as the natural parameter(i.e $|dc/ds| = 1$) then the vectors t, n, b, satisfy the Serret-Frenet equations [26], [40],

$$\begin{bmatrix} \dot{t} \\ \dot{n} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} t \\ n \\ b \end{bmatrix} \qquad \text{(III.1)}$$

where,

κ = curvature,

τ = torsion,

t = tangent,

n = normal at the point,

b = binormal at the point,

and the dot notation denotes the deravative with respect

to the natural parameter s.

The parameters κ and τ are geometric invariants and their existance and uniqueness is guaranteed by the following theorem [40],

Theorem : Let $\kappa(s)$ and $\tau(s)$ be arbitrary continious functions for $a \leq s \leq b$. Then there exists, except for position in space, one and only one curve c for which $\kappa(s)$ is the curvature, $\tau(s)$ is the torsion and s is the natural parameter.

When the curve under analysis lies in a plane, the torsion τ is equal to zero and the binormal $\dot{b}$ is constant. Thus for a curve in the x-y plane (III.1) reduces to,

$$\begin{bmatrix} \dot{t} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} 0 & \kappa \\ -\kappa & 0 \end{bmatrix} \begin{bmatrix} t \\ n \end{bmatrix} \qquad \text{(III.2)}$$

Now if θ is the angle made with respect to the x-axis by the tangent to the curve then,

$$\begin{bmatrix} \dot{t} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \qquad \text{(III.3)}$$

where i and j are unit vectors in the x and y directions respectively, differentiating (III.3) with respect to the natural parameter s yields,

$$\begin{bmatrix} \dot{t} \\ \dot{n} \end{bmatrix} = \dot{\theta} \begin{bmatrix} -\sin\theta & \cos\theta \\ -\cos\theta & -\sin\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \tag{III.4}$$

Substituting equation (III.3) in equation (III.4) yields the result,

$$\begin{bmatrix} \dot{t} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} 0 & \dot{\theta} \\ -\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} t \\ n \end{bmatrix} \tag{III.5}$$

Comparing (III.2) with (III.5)

$$\kappa = \dot{\theta} \tag{III.6}$$

In view of the different representations of the word curvature [14], [91] it is necessary to emphasize that the derivative in (III.6) is with respect to the natural parameter s and not with respect to some arbitrary distance measure. If a curve is not represented in terms of the natural parameter, but some other real valued function (say $\theta = \theta(s)$) then this transformation should be allowable. The implication is that, $\theta : I_s \to I_\theta$ is a monotonic, injective mapping of the interval I_s onto I_θ. Where I_s and I_θ are the respective domains in s and θ over which the curve is defined.

Therefore the function arctan(θ) with values between $\pm$ 1.57 radians or arccos(θ) with θ between 0 and 3.14 are not allowable changes of parameter and any

property based on these transformations may not be a property of the curve but a property of the representation. Thus if a curve or a shape has been represented in terms of the sample number 'k' and if the algorithm is unable to affect the one-to-one transformation described above, then the following fact from the Fundamental Theorem of calculus should be exploited,

$$\frac{ds}{dk} = \left| \frac{dc}{dk} \right| \qquad \text{(III.7)}$$

Using (III.7) it can be shown [40], that the magnitude of curvature can be obtained from the following relationship,

$$|\kappa| = \begin{cases} |c' \times c''| \, / \, |c'|^3 & \text{if } c' \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{(III.8)}$$

Where the symbol x denotes the vector cross-product and the symbol ' denotes differentiation with respect to k. Curvature is a vector quantity and it points in the direction of the normal to the curve.

An investigation of (III.2) suggests that curature alone is sufficient to uniquely describe curves or shapes in a plane. The problem of comparing planar shapes, is then equivalent to comparing their curvatures. This solution works well for ideal curves.

However, it is ruled out as a sole model to be used by systems, or algorithms, to analyze shapes in a manner similar to the human vision. Explicitly the following draw-backs make this solution unacceptable.

1) Curvature as defined above (proportional to the derivative of the tangent) is an operation on three adjacent points and any errors due to sampling or noise tend to make the curvature functions of any two shapes, otherwise alike for most visual purposes, unrecognizable.

2) The curvature function does not automatically render itself useful to syntactic comparision methods which is an inherent aspect of human vision.

3) Curvature is inversely proportional to the size, which implies that the plots of the logarithm of the curvature of two identical shapes of different sizes differ from each other by an additive constant. This does not seem to be a problem at first glance, however, it is a major problem when comparing partial shapes of different sizes and different number of samples. Since 'human' vision can recognize partial shapes with no extra effort, recognition of partial shapes should be a built-in part of shape analysis algorithms. The specific reason why an algorithm using only curvature as a criterion does not work well for comparing partial

shapes can be explanied as follows:

A partial shape contains a borrowed segment which belongs either to another occluding object, or to the boundary imposed at the limits of the field of view, or to itself but not to the complete shape. In other words a partial shape contains one or more segments which belong to the complete shape and some which do not. The task of comparing a partial shape to a complete one is that of locating segments on both complete and partial shapes respectively, such that their log (κ) plots differ by an additive constant. This constant can be eliminated by subtracting the means over the respective segments. The problem with this solution is that it is always possible to find a segment of length ℓ, on the partial shape, (where ℓ can be made as small as desired) such that by appropriate interpolation and resampling within the limits of resolution, the curvature of this segment can be made to match the curvature of some segment on complete shape. Therefore when no syntactic knowledge is given, or the relationship between the size and the number of samples is unknown, it is not possible to decide on the necessary length of the segments to be matched, before it could be concluded that the partial shape is a part of the complete shape. Hence without a priori knowledge

about the complete shape, matching a few segments of the partial shape to a complete shape would yield inconclusive results.

It is imperative therefore to derive some syntactic knowledge about the shape. For some shapes such as the ones shown in Fig. 2., where the peaks and valleys in the curvature correspond to the vertices or critical point of the shape, enought syntactic knowledge can be derived by finding the peaks and valleys in the curvature plot to permit a comparison.

But for shapes like the one shown in Fig. 3, sufficient information cannot be derived from the curvature plot alone to permit a comparision. Consider the curvature plot of the shape shown in Fig. 5. which is shown in Fig. 9. The plot of Fig. 9 . which was obtained by using (III.8), has only one peak corresponding to the apex of the cardioid. Also note that the curvature function obtained by using (III.8) for the cardioid before and after rotation are the identical as predicted, in sharp contrast to the differences demonstrated in Fig. 6-a and 6-b. Thus if only the peaks and valleys in the curvature were used to derive the syntactic knowledge about the shape, then the cardioid would be described by only one point, namely

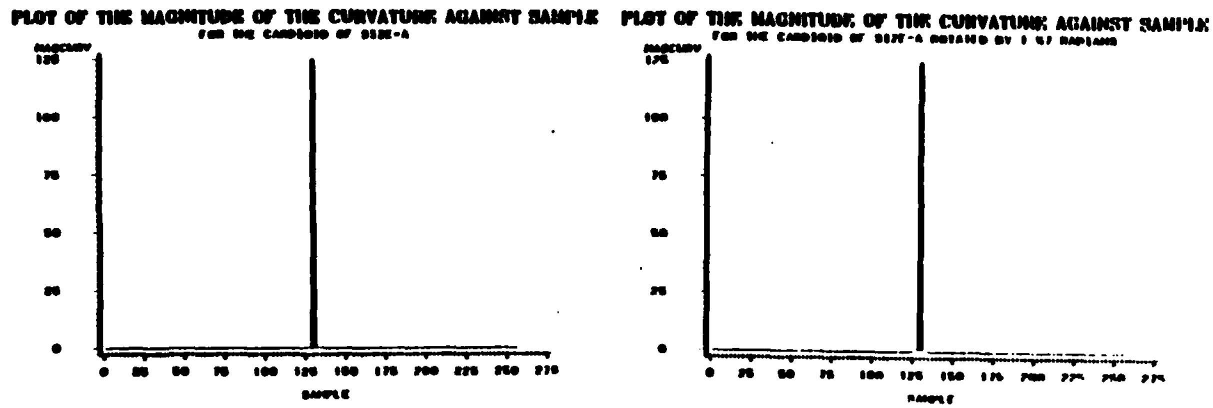


FIG. 9 CURVATURE PLOTS FOR THE CARDIOID BEFORE AND AFTER ROTATION

the apex, This is clearly insufficient. Therefore, peaks and valleys derived from curvature alone cannot be uniformly used as a single entity to represent a shape. This is not to imply that curvature is not a useful parameter, but rather to demonstrate that additional information, or attributes, are needed to fully describe the shape.

CHAPTER IV.

IV.a LINE OF SIGHT CONCEPTS.

The set of points which define a shape may be considered as a Fuzzy set in which various points are assigned to it with various degrees of membership. Defining a precise criteria on which a degree of membership can be assigned to a point on a shape is very difficult. However past reseachers have discovered [14], [19], [32], [91], that the points which should have a higher degree of membership than the rest are: 1) Points of Maxima 2) Points of Minima 3) Points of Inflection 4) Points of Intersection 5) End points of open curves 6) Points where the curvature changes sign or magnitude.

In the past shape analysis efforts, points of maxima, minima, and inflection points of curves have been extracted from the shape data without due consideration of the coordinate axes. This approach inevitably leads to errors, because these points have no meaning unless the coordinate axes are first defined. The problem with such an approach is that if a set of coordinate axes is

chosen independent of the shape under analysis then the maxima, minima, and points of inflection will not be independent of rotation, of the shape, with respect to the chosen coordinate axes. The conclusion is that the coordinate axes upon which the maxima, minima, and points of inflection are based must be dependent on the shape itself.

The next logical question is whether there should be one set of coordinate axes or many? The answer is not straight forward. Some shapes require more than one set of coordinate axes while others require only one. Before discussing the method for determining critical points, two definitions are presented which will be used in the sequel.

Definition I: A curve c is said to be in Line of Sight of a point P (LSP) if every point on the curve c can be connected to P without intersecting the curve at any other point. Otherwise the curve is said to be Not in Line of Sight of the point P (NLSP).

Examples of (LSP) curves of a point P, where P is the centroid C of the shape, which are then denoted as (LSC)-curves , are shown in Fig. 10. The concept of Line of Sight of a Point is useful not only in analysis of shapes but, it also helps in reducing the number of

computations involved in the ALS method.

Definition II: Line of Sight of a Straight line axis L: Let n be the normal projection of a curve c onto a straight line L. The curve c is then said to be in Line of Sight of Axis L (LSA) if all points from c can be mapped injectively onto n.

Examples of curves which are in Line of Sight of a single straight line Axis (LSA) are shown in Fig. 11, along with curves Not in Line of Sight of a single straight Axis(NLSA).

In the Adaptive Line of Sight method, which is discussed next, the shape is divided into a set of segments which are in line of sight of a set of axes. Dividing the shape into this set of segments is equivalent to defining the shape in terms of single valued functions. It follows from the definition of single valued function that fewer ambiguities should result, that is, the maximas, minimas, and inflection points are now determined from single valued functions rather than a multivalued function. The actual method for deterimining critical points is presented in the next section.

EXAMPLES OF SINGLY CLOSED CURVES

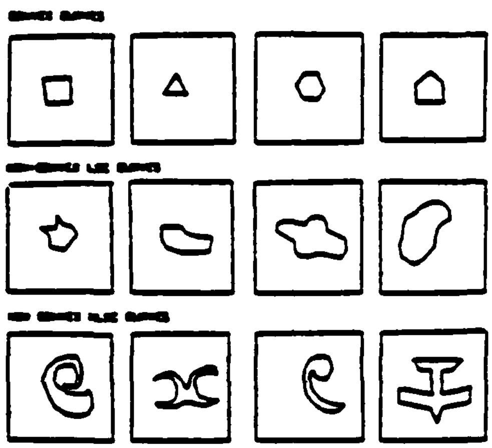


FIG.10. CURVES DEPICTING THE LINE OF SIGHT OF A POINT CONCEPT

EXAMPLES OF SINGLY OPEN CURVES

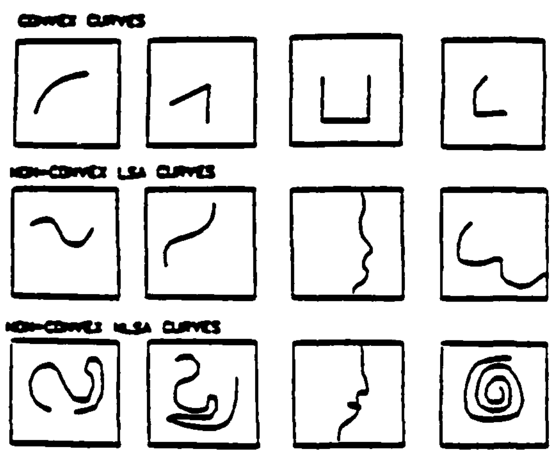


FIG. 11. CURVES DEPICTING THE LINE OF SIGHT OF A STRAIGHT LINE CONCEPT

IV.b ADAPTIVE LINE OF SIGHT METHOD.

In the Adaptive Line of Sight method the critical point determination is based on a set of coordinate axes that are dependent on the shape under consideration. As previously discussed, this will allow critical points to be determined with fewer ambiguities. The procedure is adaptive in the sense that it adapts to the shape data under consideration. This will become evident in the sequel.

The Adaptive Line of Sight method is a two pass process. In the first pass, the shape is divided into a minimum number of segments, or parts, by an appropriate set of 'critical points'. The members of this segmenting set of critical points are defined to be those points, such that all boundary points in between any two adjacent critical points have the following two properties with respect to the straight line L joining every two adjacent critical points:

1) the boundary points are on the same side of straight line L joining the adjacent critical points,

2) and the points are in line of sight of L.

From the definition of line of sight (Definition II) it is clear that this means that the boundary curve has a unique one to one projection on L. It should be emphasized that the minimum number of critical points are obtained during the first pass. This is done by an exhaustive search process that locates all points such that the above properties are satisfied. Often the minimum is not unique, in which case, the required set is obtained by summing all the minimal sets.

In the second pass, the points of maxima, minima and inflection between every two critical points are detected using the derivative of the normal distance of the point from L with respect to the distance along L. A moving average of these normal distances may be used to eliminate the effect of noise. The critical points found in pass one, the segmenting set, and the critical points found in pass two, are defined to be the members of the Fuzzy shape set with the the degree of membership depending on the number of minimal sets they are found in. A through description of both the computational and detection aspects of the Adaptive Line of Sight Algorithm along with a complete flowchart, is given in the Appendix.

The algorithm was used on the shapes shown in figures

2-a, 2-b, 3, 12, and 13. The shapes in figures 2-a and 2-b were generated by sampling the curve, obtained by linearly interpolating between the vertex points, at constant intervals of arc length. The number of samples for these shapes was arbitarily chosen to be 138 and 273 respectively. The cardioid shown in Fig. 3. which was generated by sampling the function: $r = 60 (1 + \cos\theta)$ at constant intervals of θ and not the arc length, has 256 samples. The shapes shown in figures 7 and 8 were obtained by sampling two hand sketched shapes, through a tablet, which was used as an input device, for the Tektronics 4081 graphics system. The sampling process being physical, resulted in non-uniform sampling. These shapes have 200 and 280 samples respectively. No interpolation or resampling was carried out before the data representing these shapes was input to the ALS algorithm.

The critical points obtained using the Adaptive Line of Sight of algorithm on these shapes is shown in figures 14, 15, and 16. Some of these shapes were rotated to confirm that the algorithm was independent of rotation. It can be seen from these figures that in the absence of predefined resolution the critical points tend to occur in clusters, necessitating a post processing step, such as replacing the clusters with a

PLOT OF THE PIG OF SIZE = B

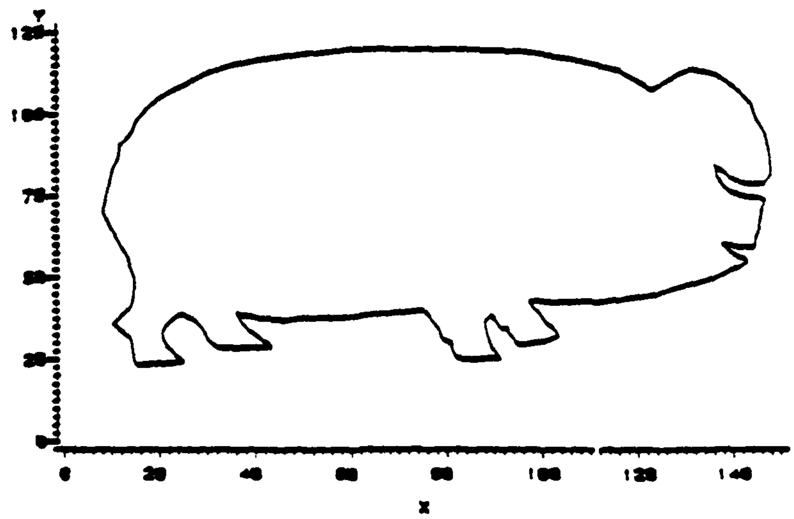


FIG. 12. THE PIG SHAPE (SAMPLED NON-UNIFORMLY).

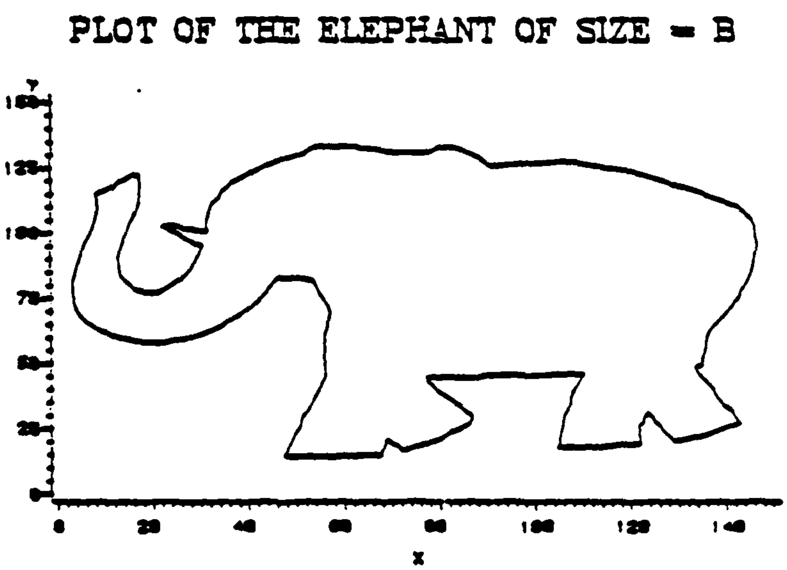


FIG. 13. THE ELEPHANT SHAPE (SAMPLED NON-UNIFORMLY)

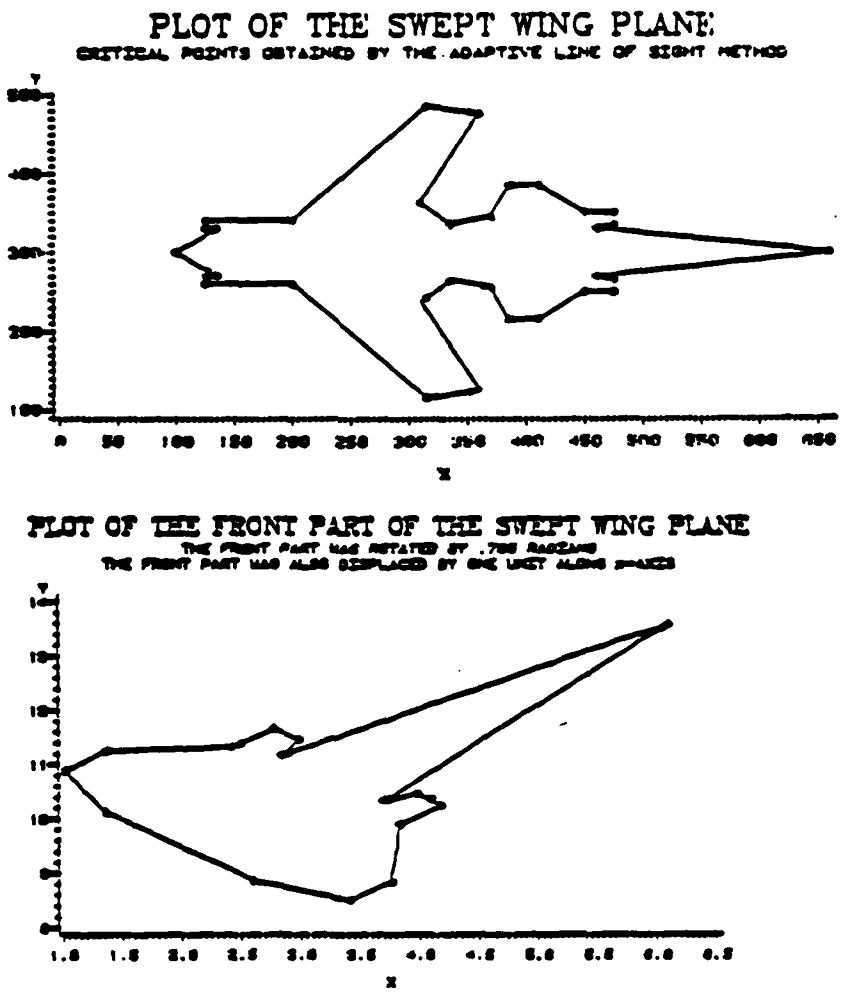


FIG. 14 . THE CRITICAL POINTS OF THE SWEPTWING PLANE SHAPES

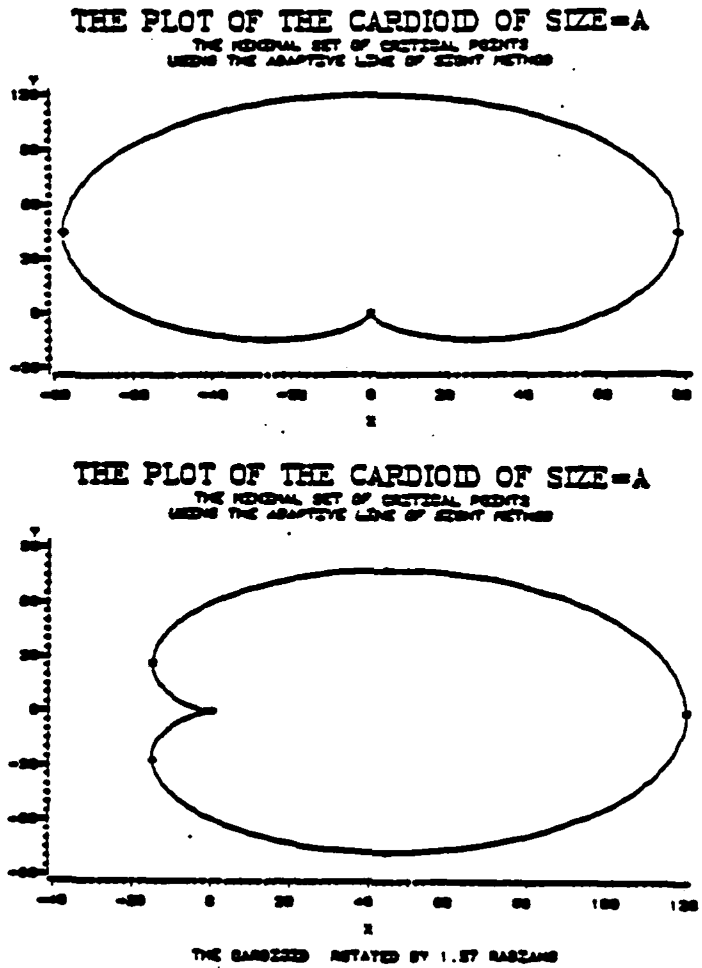


FIG.15 . THE CRITICAL POINTS OF THE CARDIOID

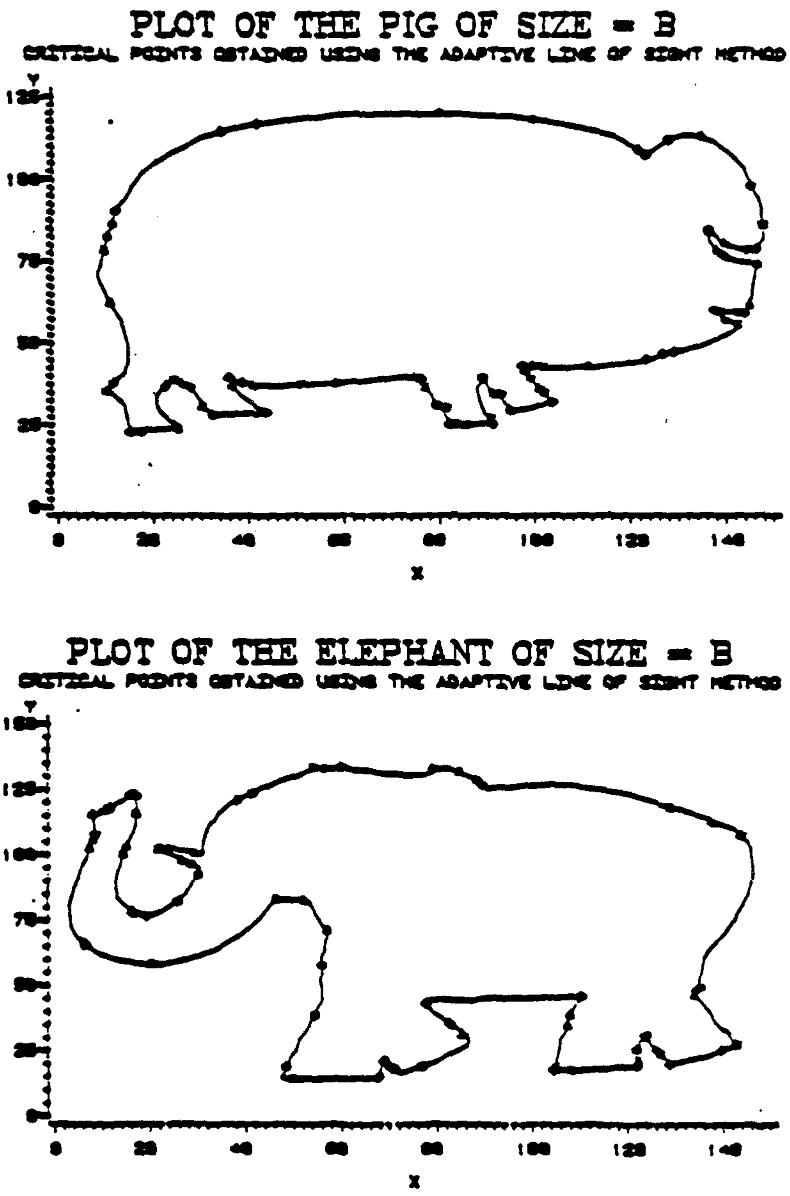


FIG. 16 . THE CRITICAL POINTS OF THE NONUNIFORMLY SAMPLED SHAPES

TYPICAL MINIMAL SETS.

single point.

Note from the shape shown that the critical points are very close to those perceived by human vision, despite of the total dissimilarity in the shapes. Also note that the post processing is done twice. First time clusters are temporarily replace by single points to count the number of critical points. The final replacement of the clusters by single points occurs when all the minimal sets have been superimposed.

In the next chapter the basic concepts of size variable, shape vector and shape space are introduced they are then used in chapter V for organizing critical points into feature vectors or subshape vectors.

CHAPTER V

BASIC SHAPE CONCEPTS

V.a MEASUREMENT VECTOR.

In this chapter several general concepts from allometric disciplines [52], are combined with some entirely new concepts concerning shapes. This combination provides the foundation for a new approach to defining shape vectors in the appropriate shape space. This new space is a combination of properties of vector spaces and is called definitively shape space.

Applying the shape space concepts to the shape analysis problem provides a basis for the recognition that the features of two or more shapes under analysis are the same. For instance, with shape space concepts it is possible to determine that a partial shape, independent of size or rotation, belongs to a more complex whole shape.

Typically in a shape analysis problem, an algorithm operates on the shape data according to a set of

criteria for the purpose of reaching a decision of some sort. Usually the decision is whether or not two or more shapes are the same. The shape analysis algorithm utilizes measurements such as curvature, and measurements between predefined points on the shape, such as length width, diameter, area, etc. Therefore, if K shapes are under analysis, and N measurements m_{nk} are made on each shape, then the result is the K measurement vectors,

$$
\begin{aligned}
M_1 &= (m_{11}\angle\alpha_{11}, m_{21}\angle\alpha_{21}, \ldots, m_{N1}\angle\alpha_{N1}) \\
&= \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
M_k &= (m_{1k}\angle\alpha_{1k}, m_{2k}\angle\alpha_{2k}, \ldots, m_{Nk}\angle\alpha_{Nk}) \\
&= \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
M_K &= (m_{1K}\angle\alpha_{1K}, m_{2K}\angle\alpha_{2K}, \ldots, m_{NK}\angle\alpha_{NK})
\end{aligned}
\qquad (V.1)
$$

where the first subscript n refers to the nth measurement between the predefined points on the k th shape. The shape or the object is represented by the second subscript. All measurements are assumed to be positive and the angle α_{nk} is the angle made by the nth measurement on the kth shape with respect to a reference direction. Any two objects will then be said to have the same shape with respect to these measurements if one vector is a scalar multiple of the

other,

$$M_g = a\, M_h \qquad \text{(V.2)}$$

where a is a scalar greater than zero.

V.b SIZE VARIABLE

The geometric significance of (V.2) is that in the N dimensional space of positive measurements all points on a straight line through the origin define the same shape. Points can be uniquely located on the positively directed line by finding the intersection of first order surfaces with the line defining the shape. The class of functions which define these surfaces are homogenous functions of order one, of the measurements, m_{nk} n=1, 2,.N. The mathematical representating for this class is,

$$\begin{aligned} \Xi\,(a m_{nk}) &= a\,\Xi\,(m_{nk}) \\ \Xi\,(m_{nk}) &\geq 0 \end{aligned} \qquad \text{(V.3)}$$

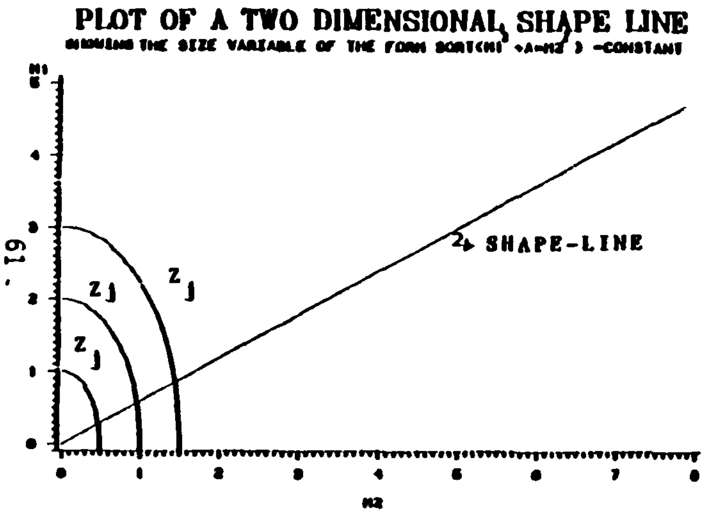
where a is scalar

and Ξ refers to a countably infinite class of functions, with members $Z_j\,(m_{nk})$.

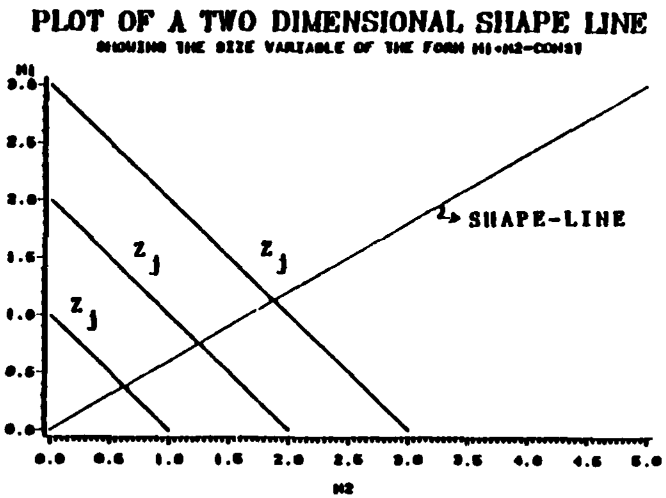
The distance from the origin to the intersection of a particular member of the class $Z_j(m_{sk})$ with the ray through origin defining the shape, is refered to as the size, scale factor or the normalization factor with respect to that member. Following Mosaiman [52] terminilogy, in the sequel $Z_j(m_{sk})$ will be refered to as the size variable. Some examples of size variable in a measurement space of two vectors are shown in Fig. 17a and b.

When comparing shapes, it is necessary to choose a size variable that is independent of the shape under comparison. A necessary requirement for the verification of this statement is the defintion of a shape vector [52], as a vector valued function of vectors, and the concept of shape space.

In all that follows it will be assumed that the measurements are always made between the centroid of the shape in question and some other fixed point on the shape.



17-a



17-b

FIG. 17. EXAMPLES OF SIZE-VARIABLE IN 2-D SHAPE SPACE.

V.c SHAPE VECTOR

The Shape vector S^{kj} is defined to be the ratio of the measurement vector M_k to the size variable Z_j.

$$S^{kj} = M_k / Z_j$$

$$S^{kj} = \left(\frac{m_{1k} \angle \alpha_{1k}}{Z_j(m_{sk})}, \frac{m_{2k} \angle \alpha_{2k}}{Z_j(m_{sk})}, \; . \; . \; . \; , \frac{m_{Nk} \angle \alpha_{Nk}}{Z_j(m_{sk})} \right) \quad (V.4)$$

It may be noted that the first superscript of S^{kj} corresponds to the object whose shape is under consideration while the second superscript corresponds to the size variable chosen from the class.

Here the points to be emphasized are that,

1) All measurements are made between pre-defined points.

2) A shape vector is defined with respect to a size variable $Z_j(m_{sk})$. Thus only shape vectors defined with respect to the same size variable can be compared.

3) If two shape vectors are equal, then the two objects have the same shape with respect to the measurements.

4) The shape vector should be independent of the size variable.

These statements merit further discussion and clarification because of their implications. Consider for example the two different measurement vectors extracted from the simple shape shown in Fig. 18. The shape is a unit square. In the first case, the measurement vector is,

$$M_1=(m_{11} , m_{21} , m_{31} , m_{41})=(1 ,1 ,1.414, 1) \tag{V.5}$$

while in the second case, the measurement vector is

$$M_2'=(m'_{12} ,m'_{22} , m'_{32} \quad m'_{42})=(1, 1, 1, 1). \tag{V.6}$$

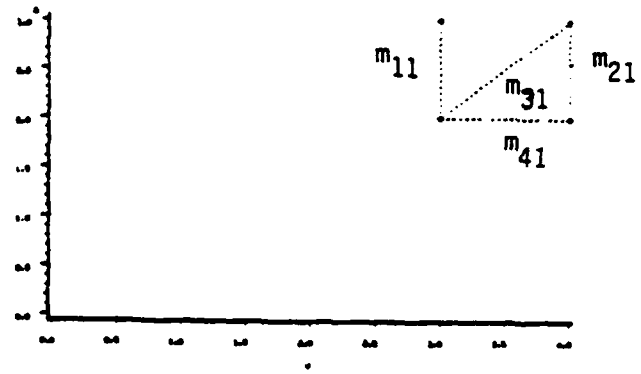
where the ' in the above equation indicates that the measurements are between a different set of feature points of the shape. Observe that in the above two equations, the measurement vectors are considered as a function of the distances only. Comparison of measurement vector as a function of the angle $\alpha_{\ell k}$ is more complex and is delayed till chapter V. Now if,

$$Z_j(m_{\ell k}) = m_{3k} \tag{V.7}$$

is chosen as the size variable then the corresponding shape vectors are,

$$S_1=(.707 , .707 , 1 , .707) \tag{V.8}$$

PLOT OF THE FEATURE POINTS ALONG WITH THE MEASUREMENTS



PLOT OF THE FEATURE POINTS ALONG WITH THE MEASUREMENTS

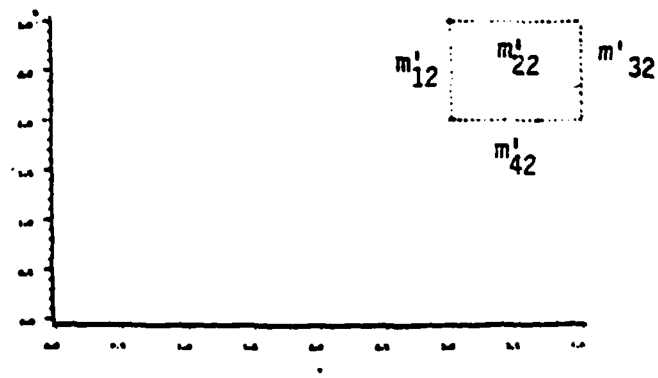


FIG. 18. MEASUREMENTS ON A SQUARE-SHAPE.

$$S_2 = (1 \quad , \quad 1 \quad , \quad 1 \quad , \quad 1) \tag{V.9}$$

Comparing these two shapes without any reference to the size variable or the points between which the measurements were made, one would conclude that the the two shapes are not the same. Thus not only has the functional form of the size variable to be the same but the measurements involved in the functional relationship have to be between the same feature points. Now consider the shapes in Fig. 19. This is the typical situation in which occlusion occurs or the shape is outside the field of view of a camera (or some other measuring device). Assume that both shape boundaries are represented by an equal number of samples and that every point on each shape boundary is defined as a feature point. Now, if the standard deviation of the data is chosen as the size variable then it is obvious that the standard deviation of the two shape boundaries are different. Therefore, this is a case where the size variable is dependent on the shape, which implies that the shape vector is dependent on the size variable. Comparision of the two shape vectors with such a size variable is bound to lead to errors. These problems can be alleviated by defining the quantities in the proper context. This can be accomplished only if the variables are defined in the proper space.

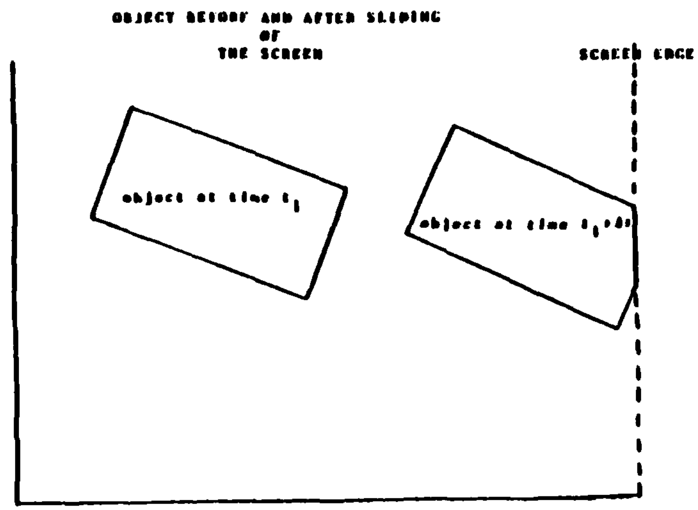


FIG. 19 . OBJECT LEAVING THE FIELD OF VIEW.

V.d SHAPE SPACE

The space is defined in terms of its properties in the usual manner and then two theorems addressing the problem of partial shapes are stated and proved. Assuming that S^{kj} (s_n^{kj}) is a shape vector consisting of well defined operations on the measurements of the shape under consideration. These measurements are the m_{nk} 's previously defined; the elements of S^{kj} (s_n^{kj}) are obtained by the following operation,

$$s_n^{kj} = m_{nk} \angle \alpha_{nk} \quad / Z_j(\quad m_{nk} \quad) \qquad (V.10)$$

The shape vector S^{kj} (s_n^{kj}) must satisfy the following properties in addition to the properties of normal Euclidean space.

PROPERTIES OF SHAPE SPACE

1) The shape vector is independent of the size variable. This implies that,

$$S^{kj} \ (\ a \ s_n^{kj}) = a \ S^{kj} \ (\ s_n^{kj}) \qquad (V.11)$$

where $n = 1, 2, \ldots\ldots, N$, $k=1, 2, \ldots\ldots K$,

and a is a scalar.

2) The shape vector is independent of translation that is,

$$S^{kj} \ (s_n^{kj} + \ s_0 \quad) = S^{kj} \ (s_n^{kj} \ ) \qquad (V.12)$$

where $n = 1, 2, \ldots\ldots N$, $k=1, 2, \ldots\ldots\ldots K$,

and s_0 is a constant vector.

3) The shape is independent of rotation.

$$S^{ki} (\alpha_{nk} + u_0) = S^{ki} (\alpha_{nk}) \qquad (V.13)$$

where n=1, 2, ...N k=1, 2, ... K,

and u_0 is a constant angle.

The vector obtained by using a set of measurements on a partial shape must still be contained in the space. Unless a size variable is found which is independent of both shapes (both the partial and the complete) it is not meaningfull to compare the shape vector in shape space. It is not possible to find a size variable which is a totally independent continious function of measurements made on both shapes. This being the case the only choice left is to split the shape into parts or subshapes and define a size variable which is piecewise continuous over these parts. Two theorems relating the subshape to the complete shape in shape space are stated and proved. These theorems are used extensively in the sequel.

V.e THEOREMS ON CONCATENATED SHAPE VECTORS.

THEOREM 1 : The vector formed by concatenating a series of shape vector is a shape vector.

Proof: It is required to show that the shape vector resulting from concatenating a series of shape vectors satisfy the three properties of shape space. Let the concatenated shape vector be

$$S^{cj} = (S^{11}\ (s_u^{11}),\ S^{22}\ (s_v^{22}),\ ..,\ S^{hh}\ (s_w^{hh}),\ ..) \tag{V.14}$$

where u = 1 ... U, v = 1 ... V, and w = 1 ... W. and the superscript c on S indicates that it is obtained by concatenating other vectors. Each element of S^{cj} (s_n^{cj}) can be represented by

$$s_n^{cj} = m_{nc}\ / Z_j\ (m_{nc}) \tag{V.15}$$

j = 1 ... J, and n = 1 ... N

where J is the total number of concatenated vectors, while N is equal to the sum of the total number of components of all the concatenated vectors.

Therefore,

$$S^{cj}\ (s_n^{cj}) = (s_1^{c1},\ s_2^{c1},\ s_3^{c1},\ ....,\ s_n^{cj},\ ....) \tag{V.16}$$

Multiplying each component of (V.14) by the scalar a

yields,

$$a\ S^{cj}(\ s_n^{cj}\ ) = (a\ S^{11}\ (\ s_u^{11}\ ), a\ S^{22}\ (\ s_v^{22}\ ), \ldots\ , a\ S^{hh}\ (\ s_w^{hh}\ )\ \ldots) \tag{V.17}$$

but from the (V.10) we have,

$$S^{hh}\ (\ a\ s_w^{hh}\ ) = a\ S^{hh}(\ s_w^{hh}\ ) \tag{V.18}$$

using (V.17) in (V.18) results in

$$a\ S^{cj}(\ s_n^{cj}\ ) = (a\ s_1^{c1}, a\ s_2^{c1}, \ldots\ \ldots, a\ s_n^{cj}, \ldots) \tag{V.19}$$

or

$$a\ S^{cj}\ (\ s_n^{cj}\ ) = S^{cj}(\ a\ s_n^{cj}\ ). \tag{V.20}$$

Equation (V.20) proves property i)

To prove that the shape vector satisfies proverty ii) it is only necessary to observe that each member of the concatenated vector is a shape vector.Therefore each satisfies:

$$S^{cj}\ (\ s_n^{cj}\ + s_0) = S^{cj}\ (\ s_n^{cj}\ ) \tag{V.21}$$

The proof for property iii) follows in a similar manner Q.E.D

Theorem2: The shape defined by a shape vector obtained by concatenating a series of shape vectors is unique if and only if the size variable and size defining each member of the set are known.

Proof: $\longrightarrow$. The surface of a homogeneous function of order one will intersect a positive directed straight

line at only one point. Since a shape is defined as a point on the shape ray, if follows that any point on this ray can be uniquely determined by its intersection of a size variable which is defined as a homogenous function over the positive quadrant.

:$\longleftarrow$. Assume otherwise. Then there is at least one $S^{hh}(s_w^{hh})$ shape vector in the concatenated set whose size variable $Z_h(m_{wh})$ can be choosen arbitarily. Then the concatenated vector under this assumption would still satisfy the properties of the shape space, i.e.,

$$a\ S^{cj}\ (\ s_n^{cj}\ )\ =\ S^{cj}\ (\ a\ \ s_n^{cj}\ ) \qquad \text{(V.22)}$$

Now the R.H.S of (V.22) can also be expanded as

$$= (\ a\ S^{11}\ ,\ a\ \ S^{22}\ ,\ \ldots\ \ ,a\ S^{hh}\ \ldots.) \qquad \text{(V.23)}$$

$$= \left(\ \frac{aM_1}{Z_1}\ ,\frac{aM_2}{Z_2}\ ,\ldots.\ \ .,\frac{a\ M_h}{Z_h}\ ..\ \right) \qquad \text{(V.24)}$$

but since Z_h can be choosen arbitrarily as long as it satisfies the definition of a size variable(V.3). Choose

$$Z_h = \quad Z_h\ (\ a\ m_{wh}\ ) \qquad \text{(V.25)}$$

substituting (V.3) and (V.25) if (V.24) the following relationship is obtained

$$a \quad S^{cj}(s_{n}^{cj}) = (\frac{a\ M_1}{Z_1} \quad \frac{, a\ M_2}{Z_2} \quad . \quad \frac{, M_h}{Z_h} \quad .. \ ) \qquad (V.26)$$

which is also equal to,

$$= \quad (a\ S^{11} \ , a \quad S^{22}, \qquad \ldots S^{hh} \ , a\ S^{h+1\ h+1} \ .. \ )$$

(V.27)

comparing (V.23) with (V.27) that,

$$a \quad S^{cj}(s_{n}^{cj}) \neq \quad S^{cj}(a \quad s_{n}^{cj})$$

which contradicts (V.1Ø)

Q.E.D

In the next chapter a method of organizing critical points into features is presented. The features are then converted into feature vectors or shape vectors using the concepts presented in this section. These shape vectors are then concatenated to form global shape vectors.

CHAPTER VI.

FEATURE SELECTION & COGNITIVE STEP

VI.a FEATURE VECTOR FORMATION PROCESS

The features of a shape are essential to defining the shape in terms of parameters that can be ultimately used by machine for decision purposes. However, the manner in which the feature defining procedure can be selected is quite variable. Since a dependable feature selection procedure is fundamental to the shape recognition problem, it is essential that this aspect of shape recognition be addressed with specificity. This point is punctuated when it is realized that, irrespective of the method of defining and detecting critical points, a cognitive algorithm is still required which examines in some sense, the critical points of the shape for the purpose of reaching a decision about some aspect of the shape. Consider, for example, the shape shown in Fig. 2-a, a shape such as this swept wing plane may have thirty to fifty critical points. The human eye makes numerous measurements, automatically and sub-

consciously, between the feature points and determines their relationship to one another. The 'most important' of these measurements combined with the relationship between them, comprise the decision set. The term " most important " is difficult to define mathematically, because it is the result of training. An unrefined cognitive procedure must therefore consider the set of all possible measurements between the critical points. Obviously this is a very large number of measurements even for numbers as modest as thirty to fifty. It is well known that this totality of measurements between critical points is not essential to the decision process.

It is necessary therefore to determine methods for acquiring the minimal set of measurements or features required for the decision process. The human, apparently places heavy weighting on features that are formed by critical points that are symmetrically opposite about an axis and features that are extracted from adjacent critical points concerning the shape. Without any prior knowledge a human can find the sets of axes about which some critical points are symmetrically placed with very little effort. However, such a task is almost insurmountable for a machine based algorithm unless it is performed at a post-cognitive level. In the absence

of noise, machine recognition (cognitive) algorithms perform reasonably well by using only features consisting of adjacent critical points.

A cognitive algorithm that utilizes measurements such as these in a continuous sequential manner would be entirely adequate if the algorithm for detecting critical points is totally immune to noise, round off, and truncation errors. For example any extra critical points that are the result of a burst of noise would prevent any continuous sequential recognition algorithm from yielding conclusive results.

One manner by which this problem can be circumvented is to divide each of the shapes under analysis into subshapes in terms of their features, and then compare the features of these subshapes and the manner in which they are related to each other. It is recalled from chapter V that the properties of shape space dictate that the measurements which define a feature must be made from the centroid of the set of critical points. It is necessary then to determine, in some manner, the minimum number of critical points that adequately define a feature. If each feature were defined in terms of only two critical points, then all features would be identical since the comparisons are

made with respect to the same shape independent size variable. Therefore, the minimum number of critical points that can form a distinguishable feature is three, and these must be adjacent. However unless the relationship between these three point features with its adjacent features is also, considered any comparison (cognitive) algorithm almost always leads to ambiguities. The reason for this is because the three point feature forms a triangle. An examination of the shape shown in Fig. 2-a shows that it contains many similar triangles.

Unfortunately the mathematics required to obtain the optimal number of critical points that should form a feature is not yet developed. Therefore, it is necessary to resort to the psychological aspects of the human recognition and decision process as well as the practical aspects such as the implementation and computational requirements. These criteria lead to the features being selected as follows:

1) Reconstruct the shape by connecting all the adjacent points by a straight line. This is called the critical shape boundary.

2) The feature F_k is then formed by including critical points, C_k and at most three adjacent critical points on each side of C_k . The critical points chosen must be in

line of sight of C_k . This means that it must be possible to draw a straight line form C_k to each of the other critical points defining the feature without intersecting the critical shape boundary. The feature obtained by this procedure cannot be defended as optimal in any mathematical sense. However, it correlates quite well with those entities that humans consider features. The features corresponding to critical points C_{12} and C_{13} are shown in Fig. 20.

A desirable improvement to the above feature defining procedure is an algorithm for deciding whether the line joining C_k to another critical point in the feature lies inside or outside the shape.

The cognitive step requires, as usual a dictionary of the features of the complete shape against which the partial shapes are to be compared. The partial shape dictionary will henceforth be referred to as the problem text. One page of the complete shape dictionary is shown in Table 1. This page contains features twenty-one and twenty-two of the swept wing plane of Fig. 21. The table includes, in addition to the feature number, the critical points of that feature along with their x and y location, the x and y location of the centroid of all the critical points contained in the feature, the size

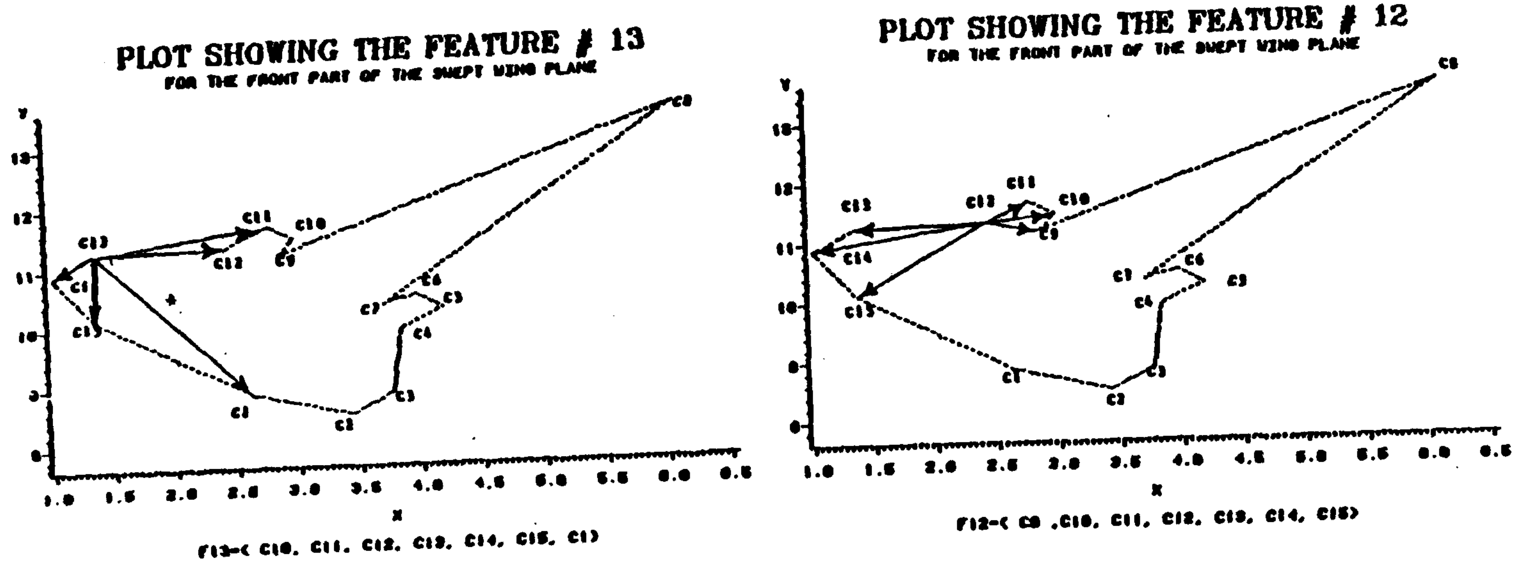


FIG. 20 . CRITICAL POINTS AND EXAMPLES OF FEATURES FOR THE FRONT PART.

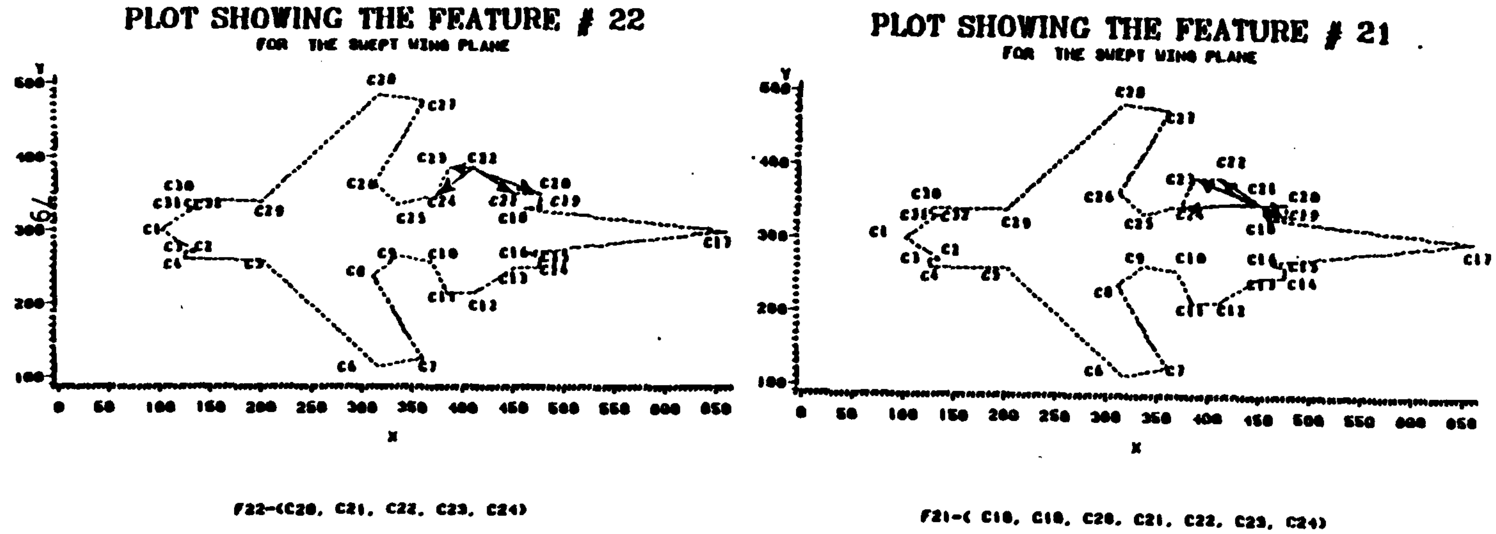


FIG. 21. CRITICAL POINTS AND EXAMPLES OF FEATURES FOR THE PLANE.

of the feature, the normalized components of the shape vector, the angle of the shape vector component and the ofsight code. The ofsight code is a binary code of length seven, associated with with the kth feature k = 1 . . .K. The nth bit of the code is equal to one if the critical point $C_{\ell-n}$ is in line of sight of the critical point C_k, where ℓ =k+4 and n =1...7.

The normalized shape vector component are defined as

$$s_n^{kj} = \text{sqrt}\ ((x_{nk} - \bar{x})^2 + (y_{nk} - \bar{y})^2\ )/\ z_j \qquad \text{(VI.1)}$$

where the size variable was chosen to have the form,

$$z_j = \sum_{n=1}^{N} \{\ (x_{nk} - \bar{x})^2 + (y_{nk} - \bar{y})^2\ \} \qquad \text{(VI.2)}$$

where the subscript k refers to the feature number, while the subscript n refers to the measurement. The angle alpha is measured with respect to a fixed reference. and is obtained by the following equation,

$$\alpha_{nk} = \begin{cases} \arctan\{(y_{nk} - \bar{y})/(x_{nk} - \bar{x})\} \\ \qquad \text{if } (x_{nk} - \bar{x}) > 0 \\ (\pi/2) + \arctan\{(y_{nk} - \bar{y})/(x_{nk} - \bar{x})\} \\ \qquad \text{if } (x_{nk} - \bar{x}) < 0 \\ (\pi/4) \quad \text{if } (x_{nk} - \bar{x}) = 0 \text{ and } (y_{nk} - \bar{y}) > 0 \\ (3\pi/4) \quad \text{if } (x_{nk} - \bar{x}) = 0 \text{ and } (y_{nk} - \bar{y}) < 0 \end{cases} \qquad \text{(VI.3)}$$

These same quantities are obtained to form a dictionary for the partial shape (problem text). Each page of both

the dictionaries begins with a feature set that is not a subset of another feature set. This feature set is defined as the uncovered feature set. The covered feature sets are arranged in the order of cardinality below the uncovered feature set on each page. The purpose of this architecture is to simplyfy the computational requirements for the cognitive step.

In general all the features in the dictionary will not be contained in the problem text. It is also true that the problem text contains features that are not present in the dictionary. This becomes apparent by examining Fig. 20. Therefore, the fact that a feature is contained in the problem text, does not imply that the partial shape is not a part of the complete shape, because it is not necessary for the partial shape to have fewer points that the whole shape. Therefore, further examination is required before a decision regarding the problem text can be made.

VI.b THE COGNITIVE PROCESS

The decision procedure consists of selecting an arbitrary word from the problem text dictionary. A

problem text word is of course a feature from the partial shape under comparison. The shape vector from the problem text is compared to the shape vector in the dictionary always starting on page one of the dictionary. The comparison continues until a match occurs.

VI.b.i THE METRIC DISTANCE

The process of matching distances is straight forward and can be achieved using linear correlation [23], metric difference, or any other metric. The following metric was used here,

$$\text{SUMABSDIF} = \sum_{n=1}^{N} \text{ABS}\,\{(s_n^{pj} - s_n^{dj})\} \qquad \text{(VI.4)}$$

VI.b.ii THE ANGULAR CORRELATION

The angle is an unreliable variable for direct comparision. Since the angle is circular variable [4], methods of circular statistics should be used for it comparison. To illustrate the issue at hand consider a feature vector whose angle vector α_{np} has the following three components.

$$\begin{bmatrix} \alpha_{1p} \\ \alpha_{2p} \\ \alpha_{3p} \end{bmatrix} = \begin{bmatrix} 0.785 \\ 1.570 \\ 5.000 \end{bmatrix} \text{ radians}$$

Now, if the feature undergoes an angular rotation of 2 radians then the corresponding angle vector α_{nd} will have the following components in the new position.

$$\begin{bmatrix} \alpha_{1d} \\ \alpha_{2d} \\ \alpha_{3d} \end{bmatrix} = \begin{bmatrix} 2.785 \\ 3.570 \\ 0.720 \end{bmatrix} \text{ radians}$$

The last component α_{3d} is equal to .72 radians as all operations are performed modulo 6.28. Note a direct comparison of the above two vectors would not be conclusive. Now, consider the angular displacemet vector δ_n^{0d} between α_{n0} and α_{nd}. It's components are,

$$\begin{bmatrix} \delta_1^{0d} \\ \delta_2^{0d} \\ \delta_3^{0d} \end{bmatrix} = \begin{bmatrix} 2.00 \\ 2.00 \\ -4.28 \end{bmatrix} \text{ radians.}$$

Observe that even though a constant was added to all the component of α_{n0} to obtain α_{nd} the difference between α_{n0} and α_{nd} is note a constant. Hence one might again be lead to believe that the two angle vectors are not the same. Using the methods of circular statistics [4], such pitfalls can be easily avoided. One way of testing whether the δ_n^{0d}'s from which the sample (components) are drawn are unidirectional or whether there is any statistical evidence of directedness is by the using the mean vector length r as a measure of correlation. The mean vector is given by,

$$q = \frac{1}{N} \sum_{n=1}^{N} e_n \qquad \text{(VI.5)}$$

where e_n's are unit vectors each pointing in the direction of the δ_n^{pd}'s. Now if the resulting vector length is denoted by R then,

$$R = \left| \sum_{n=1}^{N} e_n \right| \qquad \text{(VI.6)}$$

$$r = |q| = \frac{1}{N} R \qquad \text{(VI.7)}$$

Each of the e_n's can be expressed as,

$$e_n = \cos \delta_n^{pd} + \sin \delta_n^{pd} \qquad \text{(VI.8)}$$

substituting (VI.8) into (VI.9) the following expression for r can be obtained

$$r = \frac{1}{N} \left\{ \left(\sum_{n=1}^{N} \cos \delta_n^{pd} \right)^2 + \left(\sum_{n=1}^{N} \sin \delta_n^{pd} \right)^2 \right\}^{1/2} \qquad \text{(VI.9)}$$

The following properties make r an useful parameter for comparision of directedness of two circular vectors.

1) The value of r is independent of the zero direction.

2) r can vary between 0 and 1. If r is equal to 1 then all the circular variables (in this case the angular displacement vector components) point in the same direction, while r is equal to zero indicates that the components are randomly distributed. Values in between 0

and 1 indicate various degrees of randomness in the directedness of the components. In general for a word to match a feature the following three conditions must hold simultaneously,

1) The SUMABSDIF should be below a threshold

2) The RVALUE (value of r) should be above a threshold

3) The word and the feature should have the same ofsight code.

VI.b.iii LOCATION OF MISMATCHED POINTS

The next step is to compare the next problem text feature vector in order of cardinality, to the next feature in the dictionary and so forth. An example of this technique is given by comparing Tables 1 and 2. In this experiment a feature vector from the partial shape was selected for comparison. It should be emphasized that the feature vector is from the problem text of the partial swept wing aircraft shown in Fig. 20. The partial shape has been rotated and shifted as well as scaled to insure that any direct template matching procedure will fail. This also demonstrates that the concatenated feature vector matching procedure described

TABLE 1

A PAGE OF THE DICTIONARY OF THE SWEPT-WING PLANE

OFFSIGHT CODE	CRITICAL POINT SEQUENCE # e_n	COORDINATE X-LOCATION x_{nd}	COORDINATE Y-LOCATION y_{nd}	FEATURE VECTOR DISTANCE s_n^{dj}	FEATURE VECTOR ANGLE a_{nd}
1	18	460	330	0.122232	-.717012
1	19	475	335	0.155437	-.422854
1	20	475	350	0.142453	-.0996687
1	21	450	350	0.0607381	-.235545
1	22	410	385	0.125232	-.946161
1	23	385	385	0.186094	-.57764
1	24	370	345	0.207814	.168328

FEATURE #21
CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 558.807 ANGLE = 0.686719

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd} - \bar{x})^2 + (y_{nd} - \bar{y})^2)^{\frac{1}{2}}$

SIZE = 302.351

0	19	.	.	.	.
1	20	475	350	0.281973	-.224235
1	21	450	350	0.166587	-.385883
1	22	410	385	0.112905	-1.22203
1	23	385	385	0.191287	-.588003
1	24	370	345	0.247249	.358771
0	25	.	.	.	.

FEATURE #22
CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 553.618 ANGLE = 0.715092

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd} - \bar{x})^2 + (y_{nd} - \bar{y})^2)^{\frac{1}{2}}$

SIZE = 207.338

TABLE 2

WORDS #12 & 13 FROM THE PROBLEM-TEXT OF THE SWEPT-WING PLANE

OFFSIGHT CODE	CRITICAL POINT SEQUENCE # c_n	COORDINATE X-LOCATION x_{np}	COORDINATE Y-LOCATION y_{np}	FEATURE VECTOR DISTANCE s_n^{pj}	FEATURE VECTOR ANGLE α_{np}
1	9	2.84293	11.1716	0.122232	0.0679882
1	10	2.98446	11.4543	0.155437	0.362146
1	11	2.77241	11.6666	0.142453	0.685331
1	12	2.41872	11.3131	0.0607381	0.549455
1	13	1.35803	11.2429	0.125232	-0.161161
1	14	1.00434	10.8894	0.186094	0.20756
1	15	1.35758	10.1115	0.207814	0.933328

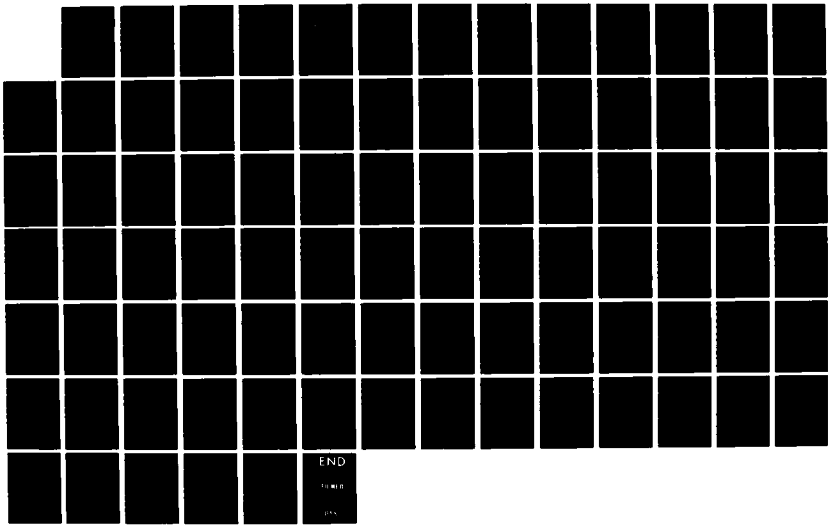
FEATURE #12
CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 11.3189 ANGLE = 1.3836
SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np} - \bar{x})^2 + (y_{np} - \bar{y})^2)^{1/2}$
SIZE = 6.04702

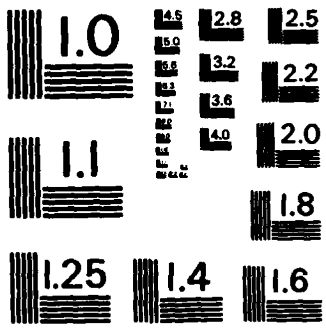
0	10	.	.	.	.
1	11	2.77241	11.6666	0.196859	0.861772
1	12	2.41872	11.3131	0.121727	0.909355
1	13	1.35803	11.2429	0.120787	-0.785796
1	14	1.00434	10.8894	0.142438	-0.227197
1	15	0.35758	10.1115	0.120787	0.785
1	1	2.62986	8.83819	0.297403	-1.2042

FEATURE #13
CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 10.8488 ANGLE = 1.39255
SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np} - \bar{x})^2 + (y_{np} - \bar{y})^2)^{1/2}$
SIZE = 6.62324

AD-A160 317 ADAPTIVE HYBRID PICTURE CODING VOLUME 2(U) ARKANSAS UNIV FAYETTEVILLE DEPT OF ELECTRICAL ENGINEERING R A JONES ET AL. 01 FEB 85 AFOSR-TR-85-0922 2/2

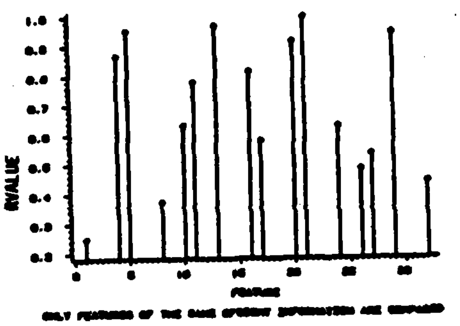
UNCLASSIFIED AFOSR-82-0351 F/G 9/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

here is independent of rotation, size and location. Feature vector twelve (word) was arbitrarily selected from the problem text. By comparing tables 1 and 2 From of the plot the RVALUE and the SUMABSDIF shown in Fig. 22-a and Fig. 22-b it is apparent that word 12 matched feature vector 21 of the dictionary. It should be noted that this match occurs even though the location of the critical points and the centroid of the feature of the partial shape are different from those same quantities for the whole shape because of the rotation and shift. In this way a correspondence table is then established between the critical points of the features in the dictionary to the critical points of the word in the problem text. The next step is to proceed in sequential order to the next word in the problem text e.g. word 13 (n+1) which is sequentially next to word 12 (n) of the problem text does not match feature vector 22 (m+1) of the dictionary. However since feature vector 22 is on the same page as feature vector 21 (m) of the dictionary , word 13 is mismatched to feature 22 because it contains a critical point which is not contained in word 12. An examination of tables 1 and 2 indicates that the critical point C_1 is contained in word 13 but not in word 12.



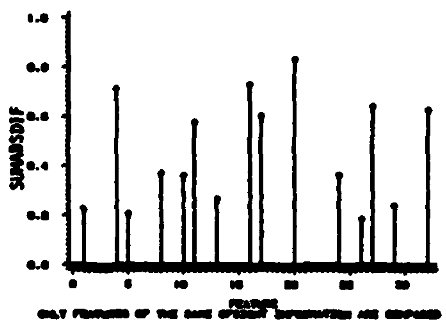


FIG. 22 . PLOT OF ANGULAR CORRELATION AND THE DISTANCE METRIC

FOR WORD 12 OF THE PROBLEM TEXT OF THE PLANE AGAINST FEATURES IN THE DICTIONARY.

The mismatched critical point is first compared to the correspondence table. If it is not found in the correspondence table then it is stored in a mismatch table. At any latter stage a mismatched critical point is erased from the mismatched table if some word containing the mismatched critical point matches some feature of the dictionary.

In general if word n matches feature m then it is expected that word (n +1) will match feature (m + 1). If feature (m+1) is on the same page as feature m then it is easy to isolate the mismatched point as in the above example. If feature m+1 is not on the same page as feature m or the concept of pages is not used then in order to isolate the mismatched critical point then the feature m+1 and the word n+1 have to be revised into concatened shape vectors with each subshape vector of three measurements. A revised feature vector for feature 22 is shown in Table 3. While the revised words for word 13 are shown Table 4. A comparison of the above two tables again isolates C_1 as the mismatched critical point. It must be noted again from these tables that angle being a multivalued function cannot be relied upon as a variable that can be used in the direct comparison process.

TABLE 3

REVISED FEATURE VECTOR #22 OF THE DICTIONARY

CRITICAL POINT SEQUENCE #	COORDINATE X-LOCATION	COORDINATE Y-LOCATION	FEATURE VECTOR DISTANCE	FEATURE VECTOR ANGLE
c_n	$x_{nd'}$	$y_{nd'}$	β'^{j}_{n}	$\alpha_{nd'}$
20	475	350	0.370213	-0.370891
21	450	350	0.145986	-1.1659
22	410	385	0.483801	2.552

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 373.435 ANGLE = 0.682458

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd'} - \bar{x})^2 + (y_{nd'} - \bar{y})^2)^{1/2}$

SIZE = 86.9464

21	450	350	0.483801	-0.370891
22	410	385	0.145986	-1.1659
23	385	385	0.370213	2.552

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 358.214 ANGLE = .732593

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd'} - \bar{x})^2 + (y_{nd'} - \bar{y})^2)^{1/2}$

SIZE = 86.9464

22	410	385	0.355588	0.331635
23	385	385	0.192090	0.91418
24	370	345	0.452316	4.11851

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 537.53 ANGLE = .763472

SIZE VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd'} - \bar{x})^2 + (y_{nd'} - \bar{y})^2)^{1/2}$

SIZE = 71.5451

TABLE 4

REVISED FEATURE #13 OF THE PROBLEM TEXT

CRITICAL POINT SEQUENCE #	COORDINATE X-LOCATION	COORDINATE Y-LOCATION	FEATURE VECTOR DISTANCE	FEATURE VECTOR ANGLE
e_n	x_{np}'	y_{np}'	$\rho_n^{i,j}$	θ_{np}'
11	2.7724	11.6666	0.370213	-0.370891
12	2.4187	11.3131	0.145986	-1.1659
13	1.3980	11.2629	0.483801	2.552

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 11.4687 ANGLE = 0.682438

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np} - \bar{x})^2 + (y_{np} - \bar{y})^2)^{1/2}$

12	2.4187	11.3131	0.483801	-0.388003
13	1.3980	11.2429	0.145986	1.9741
14	1.0043	10.8894	0.370218	2.76911

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 11.1643 ANGLE = 0.732593

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np} - \bar{x})^2 + (y_{np} - \bar{y})^2)^{1/2}$

13	1.3980	11.2429	0.335348	0.331635
14	1.0043	10.8894	0.192098	1.81418
15	1.3575	10.1115	0.452314	4.10851

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE = 10.7906 ANGLE = 0.783472

SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np}' - \bar{x})^2 + (y_{np}' - \bar{y})^2)^{1/2}$

SIZE = 1.4309

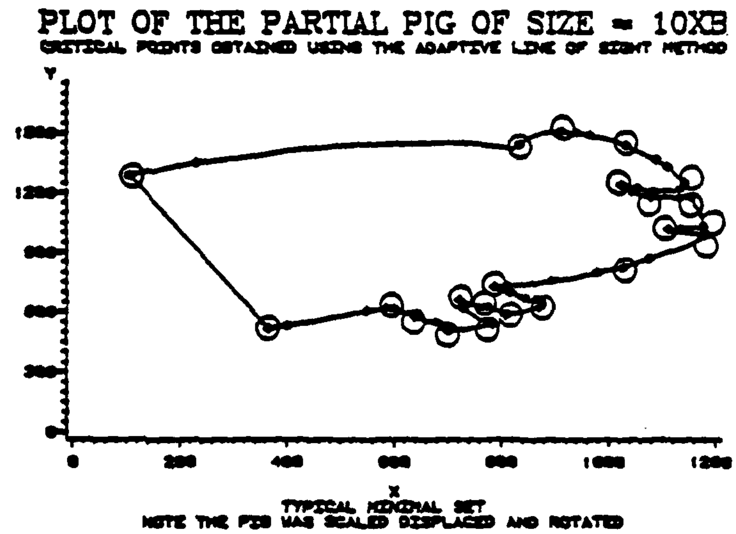
16	1.0043	10.8894	0.38763	1.390122
15	1.3575	10.1115	0.117218	1.86666
1	2.6298	8.8382	0.495153	4.64272

CENTROID LOCATION RELATIVE TO REFERENCE
DISTANCE 9.96851 ANGLE = 0.719143

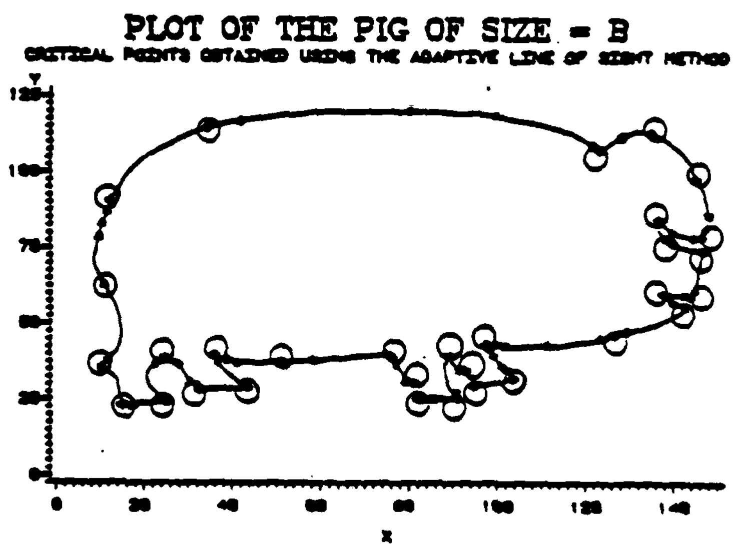
SIZE-VARIABLE $z_j = \sum_{n=1}^{N} ((x_{np}' - \bar{x})^2 + (y_{np}' - \bar{y})^2)^{1/2}$

SIZE = 2.96893

The usefulness of the Fuzzy shape concept is demonstrated by the results of the ALS algorithm on the pig of size B and the partial pig of size 10xB. More than one minimal set can be obtained by starting at various points on the back of the pig or the earlobe of the pig. However, the points which are found most often in minimal sets have been enclosed by circles and are shown in Fig 23-a and 23-b. These clusters or points of the highest degree of membership are retained by the algorithm while the others are discarded. The effect of postprocessing on the clusters of highest degree of membership gives the final minimal set. The final minimal set for the pig and the partial pig are shown in Figures 24-a and 24-b respectively. Word 21 of the partial pig is listed in table 6. The plot of the RVALUE and the SUMABSDIF of word 21 when compared with features with the same ofsight code in the dictionary is given if Fig 25-a and 25-b. From these plots it is clear that word 21 matched feature 31. The details of feature 31 are listed in table 5.

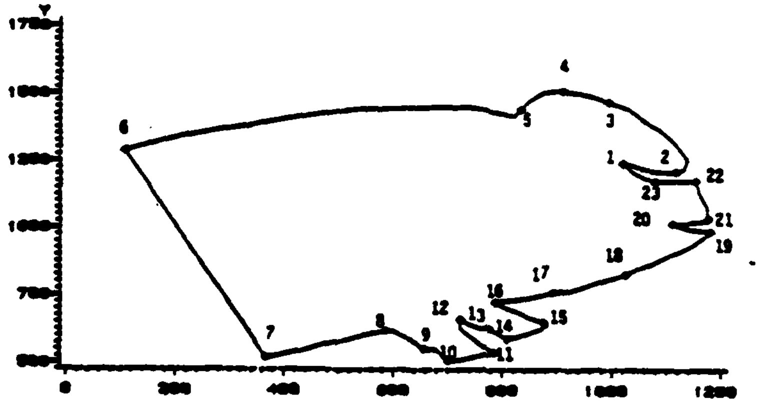


23-a

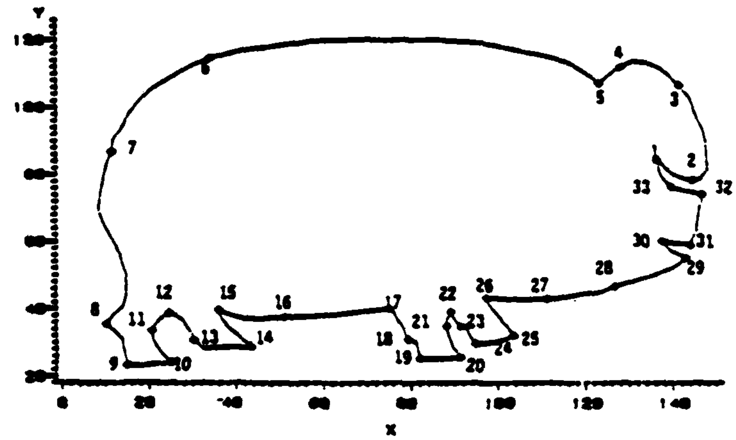


23-b

FIG. 23 . THE POINTS OF HIGHEST DEGREE OF MEMBERSHIP IN A MINIMAL SET.



24-a



24-b

FIG. 24 . THE FINAL MINIMAL SETS FOR THE COMPLETE AND THE PARTIAL PIG.

A PAGE OF THE DICTIONARY OF THE PIG.

TABLE 5

	CRITICAL POINT	COORDINATE		FEATURE VECTOR	
OFFSIGHT CODE	SEQUENCE NO	X-LOCATION	Y-LOCATION	DISTANCE	ANGLE
	e_n	x_{nd}	y_{nd}	e_n^{dj}	e_{nd}
0	28	.	.	.	.
1	29	142.802	55.0989	0.203466	-1.4372
1	30	137.361	60.2437	0.135065	1.14005
1	31	143.852	58.9716	0.148648	-1.27583
1	32	146.515	74.0838	0.122912	0.817533
1	33	139.493	76.502	0.128957	-1.38617
1	1	136.192	84.544	0.260973	-1.282

FEATURE #31

CENTROID LOCATION RELATIVE TO REFERENCE

DISTANCE 1650.99 ANGLE = 0.870066

SIZE VARIABLE $z_j = \sum_{n=1}^{N} ((x_{nd} - \bar{x})^2 + (y_{nd} - \bar{y})^2)^{\frac{1}{2}}$

A PAGE OF THE PROBLEM TEXT OF FIG.

TABLE 6

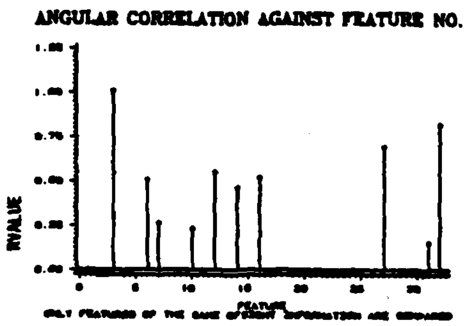
OFFSIGHT CODE	SEQUENCE NO	X-LOCATION	Y-LOCATION	DISTANCE	ANGLE
	c_n	x_{np}	y_{np}	e_n^{pj}	e_{np}
0	18	.	.	.	.
1	19	1179.47	987.51	0.208825	-1.10959
1	20	1111.23	1018.35	0.139664	0.47611
1	21	1176.13	0138.38	0.139058	-0.88802
1	22	0052.5	1179.07	0.1282	-1.04344
1	1	1020.75	1244.18	0.26216	-0.943232

FEATURE #21

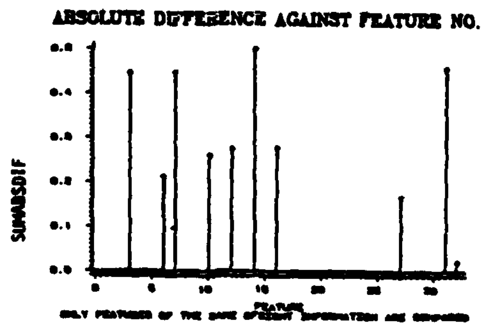
CENTROID LOCATION RELATIVE TO REFERENCE

DISTANCE = 1650.99   ANGLE = 0.870066

SIZE VARIABLE $Z_j = \sum_{n=1}^{N} ((x_{np} - \bar{x})^2 + (y_{np} - \bar{y})^2)^{\frac{1}{2}}$



25-a



25-b

FIG. 25. PLOT OF THE ANGULAR CORRELATION AND THE DISTANCE METRIC.
FOR WORD 21 OF THE PROBLEM TEXT OF THE PIG AGAINST FEATURES IN THE DICTIONARY

CHAPTER VII.

CONCLUSION AND DISCUSSION

VII.a CONCLUSION

A system for the recognition of shapes which is in some sense similar to human vision system is described. The system is called the partial shape recognition system since it is capable of recognizing shapes based on comparison of properties of parts of a shape rather than only the global properties. Though not all aspects involved in implementing the system were discussed, some of the necessary aspects were detailed.

A new concept of treating shapes as vectors in shape space is introduced and described. Also two theorems relating to the process of comparing partial shapes to the complete shape were stated and proved.

A new procedure of determining (the correspondence tokens) the critical points of a shape is described. The procedure is named the Adaptive Line of Sight method. In the Adaptive Line of Sight method the critical point

determination is based on a set of coordinate axes that are dependent on the shape being examined. Examples were given that demonstrate that the procedure produces critical points that are independent of rotation, size, displacement and correspond closely to those produced by normal human cognitive process.

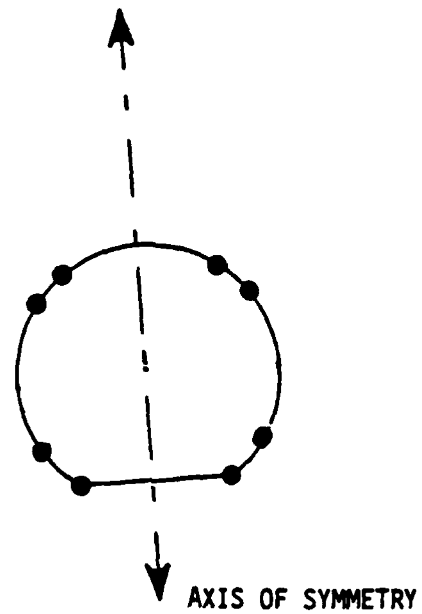
It was shown that the critical points could be organized to form feature vectors using the Line of Sight of a Point concept. A technique for comparing the feature vectors of a set of shapes is described. The comparison procedure is based on syntactic technique which will point whether one shape is part of a more complex whole shape, or whether the shapes are totally dissimilar.

VII.b DISCUSSION

VII.b.i CAN COMPARISON OF SHAPES BE BASED ON CRITICAL POINTS ALONE ?

Examples presented in the earlier sections showed that the shape recognition technique based on comparing

critical points of the highest degree of membership was sufficient and gave good results. There are instances however, when a match based on comparison of critical points alone is insufficient. A simple example is that of an ellipse. The critical points of the highest degree of membership of the ellipse shape are the points where the ellipse intersects it minor and major axes. These critical points are also the critical points of a polygon (which looks different than an ellipse) formed by joining the adjacent critical points by a straight line. A comparison based on critical points alone would not be able to make out the difference between the polygon and the ellipse. *In general* the shape represented by the critical shape boundary and the shape boundary are indistinguishable. Another example is that of the circle and, a circular arc which is greater than a semi-circle but less than the circle. For the circle all points will be determined to be equally important. For the circular arc all points excluding the points shown in Fig. 26. will be determined to have the same degree of membership. The points shown on the circular arc have a higher degree of membership than the rest. Here a comparison based on matching critical points of the highest degree of membership would lead to wrong results. In such instances it is proposed to match the



NOTE: ALL POINTS ON THE ARC ARE CRITICAL POINTS BY THE ADAPTIVE LINE OF SIGHT METHOD; HOWEVER THE POINTS WITH THE HIGHEST DEGREE OF MEMBERSHIP HAVE BEEN ENCLOSED BY CIRCULAR DOTS

FIG. 26. A CASE FOR RELOCATION OF CRITICAL POINTS ?

shapes based on the space invariant properties of the segments (interpolated and resampled if necessary) between critical points. Figure 1. thus shows a block "length of the segments to be matched" which is also loaded into the shape data table.

In this paper the actual procedures for matching segments were not completely addressed. The above examples suffice to point out that comparison based on curvature of segments, or other space invariant properties of the segments is also an integral part of the partial shape recognition procedure.

VII.b.ii REASSIGNMENT AND RELOCATION OF CRITICAL POINTS BASED ON SYMMETRY

There is one more problem associated with the case of the circular arc. It is true that most humans assign the highest degree of membership to the end points of the arc but it is also true that most of them do not assign similar degree of membership to the remaining points on the arc. Specifically it is usually assumed that the critical points found relative to the axis of symmetry are more important than the rest. It can be argued that

the determination of such an axis of symmetry occurs at some post-cognitive level. In other words it occurs only when parts of the shape are 'compared' with itself. The comparison signified by the word 'compared' in the previous sentence occurs before the axis of symmetry has been located. It is this kind of comparison which has been addressed in this paper. This ofcourse excludes the redefinition of critical points and ressignment of their degrees of membership based on the feed-back of the results of comparison of the shape with a copy of it-self (see Fig. 1.), and falls outside the dotted line shown in Fig. 1.

VII.b.iii HOW TO SET THE THRESHOLDS.

An important question associated with the above method is that of the thresholds in the algorithm. More features can be determined at lower thresholds, while many features can be eliminated at higher thresholds. Resampling the curve at uniform arc length is a necessary first step in the determination of the thresholds; even though we did not resample some of the shape at uniform arc length to get a feel of the

robustness of the algorithm. This is the reason why some of the critical points were missed near the leg of the pig shape in Fig. 16. In general the exact threshold will depend on

1) Resolution of the Machine (a machine may have variable range of thresholds)

2) Knowledge from the data base (see Fig. 1.)

3) Feedback from the output table (see Fig. 1.)

It is not clear how the thresholds can be adjusted using the feedback, however there appear to be at least two factors which seem to influence this aspect, namely,

1) Error or Noise

2) Number of 'small features'

The former means an estimate of the error between the best fit polynomial and the linearly interpolated curve. The latter comes from a general observation that humans tend to ignore (average out) features if they are smaller than the overall global size, and if there are a large number of them. However if there are only a couple of small features they often become a point of 'focus'. Thus small features cannot be generally ignored.

VII.c FUTURE WORK

It should be very clear from the problem statement in the introduction that only the problems in the area enclosed by the dotted line in Fig. 1. have been addressed in this dissertation. A lot of work still needs to be done in the future before the complete implementation of the system shown in Fig. 1. becomes a reality. Some of the important topics that need to worked on are listed as follows:

1) Edges of objects or shapes, have been successfully detected in images. For a brief summary of the methods used to detect edges the reader is advised to refer to [16]. The problem of transforming the edges into a sequence of points defining a shape has also been attempted [10], [16]. These methods work reasonably well on ideal data specially when the points are located at nearly uniform intervals (e.g. eight connected neighbourhood). However these methods fail to work on general edge data (i.e. data defining the edge of the object), which is normally obtained using edge detectors [16], on actual 'real life' images. So work needs to be done either to alter these methods so that they can adapt to the edge data, or the methods of detecting

edges need to be modified so as to produce output data which is close to ideal.

2) The question of thresholds which has already been addressed also needs to be worked on. Explict standards defining the range of resolution of the machine need to to laid.

3) Most of the state of the art work [30], [54], [68], [71], [88], pertaining to the interpretation process described in Chapter I. deals with the extraction of motion information from changes in the projection of the object on a plane. Practically no actual work has been done which exploits the relationship of the object with respect to the background towards this end.

4) Though some guide lines can be drawn from the field of psychology for the implementation of a workable knowledge data base shown in Fig. 1, the actual implementation of a general of a data base comparable to the human vision system is still a far cry form reality.

REFERENCES

1) Ables, J. B., "Maximum Entropy Spectral Analysis," Proc. Symp. on the Collection and Analysis of Astrophysical Data, 1972, pp. 23-33.

2) Andersen, N., "On the Calculation of Filter Coefficients for Maximum Entropy Spectral Analysis," Geophysics, Vol. 39, No. 1, Feb. 1974, pp. 69-72.

3) Badler, N., "Three Dimensional Motion from 2-D Picture Sequences," Second International Joint Conferences on Pattern Recognition, Copenhagen. Denmark, August 1974, pp. 157-161.

4) Batschelet, E., Circular Statistics in Biology, Academic Press, 1981.

5) Bjorklund, M. C., and Pavlidis, T., "Global Shape Analysis by k-Syntactic Similarity," IEEE Trans., Vol. PAMI-3, No. 2, March 1981, pp. 144-155.

6) Blackman, B., and Tuckey, J. W., "The Measurement of Power Spectra from the Point of View of Communication Engineering," Dover Press, NY. 1959.

7) Bribiesca, E., "Arithmetic Operations among Shapes using Shape Numbers," Pattern Recognition, Vol. 13, No. 2, 1981, pp. 123-138.

8) Burg, J. P., "Maximum Entropy Spectral Analysis," Proceedings of the 37th meeting of the Society of Exploration Geophysicists, 1969, pp. 34-41.

9) Buzo, A., Gray, A. H., Gray, R. M., and Markel, J. D., "Speech Coding Based Upon Vector Quantitization," IEEE Trans., Vol. ASSP-28, No. 5, Oct. 1980, pp. 62-573.

10) Chidananda, Gowda K., and Krishna, G., "The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood," IEEE Trans., Vol. IT-25, No. 4, July 1979, pp. 488-491.

11) Cornsweet, T., Visual Perception, Academic Press, Inc., N.Y., 1970.

12) Crow, C. Franklin, "Towards More Complicated Computer Imagery," Vol. 5, No. 2-4, 1980, Pergamon Press, Ltd., N.Y.

13) Csiszar, I., "I-Divergence Geometry of Probability Distributions and Minimization Problems," Annals of Prob., Vol. 3, No. 1., 1975, pp. 146-158.

14) Davis, L. S., "Understanding Shapes: Angles and Sides," IEEE Trans., Vol. C26, No. 3, March 1977, pp. 236-242.

15) Dodd, G., and Rossol Lothar, (ed.), Computer Vision & Sensor Based Robots, Plenum Press, N.Y., 1979.

16) Duda, R. O. and Hart, P. E., Pattern Classification and Scene Analysis. John Wiley and Sons, 1973.

17) Elliot, H., Cooper, D. B., Cohen, F. S., and Symoskek, P. F., "Implementation, Interpretation, and Analysis of Sub-optimal Boundary Finding Algorithm," IEEE Trans., Vol. PAMI-4, No. 2, March 1982, pp. 167-191.

18) Espiau, B., and Catros, J. Y., "Use of Optical Reflectance Sensors in Robotics Application," Correspondence, IEEE Trans. System Man and Cybernetics, December 1980, Vol. SMC-10, Number 12, pp. 903-912.

19) Freeman, H., "Shape Description via Critical Points," Pattern Recognition, Vol. 10, pp. 159-166.

20) Fu, K. S., Syntactic Methods in Pattern Recognition. Academic Press, New York, 1974.

21) Granlund, G. H., "Fourier Preprocessing for Hand Printed Character Recognition," IEEE Trans., Vol. C-21, Feb. 1972, pp. 195-201.

22) Gray, R. M., Buzo, A., Gray, H. A., and Matsuyama, Y., "Distortion Measures for Speech Processing," IEEE Trans., Vol. ASSP-28, No. 4, August 1980, pp. 367-375.

23) Hall, E. L., Davies, A., and Casey, M. E., "The Selection of Critical Subsets for Signal, Image, and Scene Matching," IEEE Trans., Vol. PAMI-2, No. 4, July 1980, pp. 313-322.

24) Hason, A., and Riseman, E., "Design of VISIONS: Segmentation & Interpretation of Images," Joint Workshop on Pattern-Recognition & Artificial Intelligence, IEEE Press, 1976, pp. 135-141.

25) Harmon, L. D., Khan, M. K., Lasch, R., and Ramis, R. F., "Machine Identification of Human Faces," Pattern Recognition, Vol. 13, No. 2, 1981, pp. 97-122.

26) Hermann, R., Differential Geometry and the Calculus of Variations. Academic Press, New York, N.Y., 1968.

27) Hopcroft, J. E., and Ullaman, J. D. Formal Languages & Their Relation to Automata, Addison-Wesley, Reading, MA, 1969.

28) Jacobus, C. J., Chien, R. T., and Selander, J. M., "Motion Detection and Analysis by Matching Graphics of Intermediate Level Primitive," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 495-510.

29) Jain, R., Martin, W. N., and Aggarwal, J. K., "Segmentatation through the Detection of Changes due to Motion," Computer Graphics & Image Processing, 1979, 11, pp. 13-34.

30) Jain, R., "Extraction of Motion Information From Peripheral Processes," IEEE Trans., Vol. PAMI-3, No. 5, September 1981, pp. 489-503.

31) Jaynes, E. T., "Information Theory and Statistical Mechanics (Paper I & II)," Phys. Rev. 106, 620:108, 171.

32) Johnston, E., and Rosenfeld, A. "Angle Detection on Digital Curves," IEEE Trans., Vol. C-22, 1973, pp. 875-878.

33) Jones, R. A., and Rashid, H. U., "Residual Recursive Displacement Estimation," IEEE Computer Society Conference, Pattern Recognition and Image Processing, August 1981, pp. 508-511.

34) Jones, R. A., and Tejwani, Y. J., "Determination of Critical Points on Planar Shapes Using the Adaptive Line of Sight Method." Submitted to Pattern Recognition.

35) Judea, P., "An Application of Rate Distortion Theory to Pattern Recognition," Pattern Recognition, Vol. 18, pp. 11-22.

36) Judea, P., "On Connection Between the Complexity of Inferred Models," Int. J. General Systems, 1978, Vol. 4, pp. 255-264.

37) Kashyap, R. L., and Chellappa, R., "Stochastic Models for Closed Boundary Analysis. Representation and Construction," IEEE Trans., Vol. IT-27, No. 5, September 1981, pp. 627-637.

38) Kim, C. E., "On the Cellular Convexity of Complexes," IEEE Trans., Vol. PAMI-3, 1981, pp. 617-625.

39) Kim, C. E., and Rosenfeld, A., "Digital Straight Lines and Convexity of Digital Regions," IEEE Trans., Vol. PAMI-4, No. 2, March 1982, pp. 149-153.

40) Klingenberg, W. A Course in Differential Geometry. Springer-Verlag, N.Y., 1978.

41) Kullback, S., Information Theory and Statistics, Dover Press, NY, 1968.

42) Lacoss, R. T., "Data Adaptive Spectral Analysis Methods," Geophysics, Vol. 36, No. 4, August 1971, pp. 661-675.

43) Lattiuati, V., and Lemonie, D., "Closed Contour Extraction Application to Meteorogical Pictures," Pattern Recognition, Vol. 15, No. 3, 1982, pp. 145-152.

44) Lerner, E. J., "Computers That See," IEEE Spectrum, October 1980, pp. 28-33, Vol. 17, No. 10.

45) Lim, J. S., and Malik, N. A., "A New Algorithm for Two Dimensional Maximum Entropy Power Spectrum Estimation," IEEE Trans., Vol. ASSP-29, No. 3, January 1981, pp. 401-412.

46) Macina, N. A., "Maximum Entropy Modeling for Identification and Detection of Certain Classes of Dynamic Events," RCA Review, Vol. 42, March 1981, pp. 85-110.

47) McDonough, R. N., "Maximum-Entropy Spatial Processing of Array Data," Geophysics, Vol. 39, No. 6, December 1974, pp. 843-851.

48) Moellering, H., and Rayner, J. N., "The Harmonic Analysis of Spatial Shapes Using Dual Axis Fourier Shape Analysis (DAFSA)," Geographic Analysis, Vol. 13, No. 1, January 1981, pp. 64-77.

49) Mokhoul, J., "On the Eigenvectors of Symmetric Toepliz Matrix," IEEE Trans., Vol. ASSP-29, No. 4, August 1981, pp. 869-872.

50) Moore, D. J. H., "A Configuration Theory of Visual Perception, Int. J. Man-Machine Studies, 1975, pp. 449-504.

51) Mosimann, J. E., "Size Allometry: Size and Shape Variables with Characterizations of the Log-Normal and Generalized Gamma Distribution," J. ASA. June 1970, Vol. 65, pp. 930-945.

52) Mosimann, J. E., "Statistical Problems of Shape and Size. I. Statistical Distribution in Scientific Work, Vol. 2, 1975, pp. 219-239.

53) Mosimann, J. E. "Statistical Problems of Size and Shape. II. Statistical Distribution in Scientific Work, Vol. 2, 1975, pp. 219-239.

54) Nagel, H., "Analysis Technique for Image Sequences," Proceedings of the Fourth International Joint Conference on Pattern Recognition, IEEE, Inc. 1979, pp. 186-213.

55) Netravali, A. N., and Robbins, J. D., "Motion-Compensated Television Coding: Part-1," The Bell System Technical Journal, Vol. 58, No. 3, March 1979, pp. 631-669.

56) Neumann, B., "Exploiting Image Formation Knowledge for Motion Analysis," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 550-554.

57) Nevins, A. J., "Region Extraction from Complex Shapes," IEEE Trans., Vol. PAMI-4, No. 5, Sept. 1982, pp. 500-511.

58) Papoulis, A., "Maximum Entropy and Spectral Estimation: A Review," IEEE Trans., Vol. ASSP-29, No. 6, December 1981, pp. 1176-1186.

59) Parzen, E., "Multiple Time Series Modeling," in Krishnaiah, P. R. (ed.), Multivariate Analysis II. Academic Press, 1969, pp. 389-409.

60) Pavilidis, T., Structural Pattern Recognition. Springer-Verlag, N.Y., 1977.

61) Pavilidis, T., "Algorithms for Shape Analysis of Contours and Waveforms," IEEE Trans., Vol. PAMI-2, No. 4, July 1981, pp. 301-312.

62) Peacock, R., and Mylopoulous, J., "A Region-based Formalism for Picture Processing," Pattern Recognition, Vol. 13, No. 6, 1981, pp. 399-415.

63) Persoon, E., and King-Sun Fu, "Shape Discrimination Using Fourier Descriptors," IEEE Trans., Vol. SMC-7, No. 3, March 1977, pp. 170-180.

64) Proffit, D., "Normalization of Discrete Planar Objects," Pattern Recognition, Vol. 15, No. 3, pp. 137-143.

65) Rashid, R. F., "Towards a System for the Interpretation of Moving Light Display," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 554-561.

66) Richard, C. W., Jr., and Hemami, H., "Identification of Three Dimensional Objects using Fourier Descriptors of the Boundary Curve," IEEE Trans., Vol. SMC-4, July 1974, pp. 371-378.

67) Roach, J. W., and Aggarwal, J. K., "Determining the Movement of Objects from a Sequence of Images," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 554-561.

68) Roach, J. W., and Aggarwal, J. K., "Computer Tracking of Objects Moving in Space," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-1, No. 2, April 1979, pp. 127-135.

69) Rosenberg, B., "The Analysis of Convex Blobs," Computer Graphics and Image Processing," Vol. 1, 1972, pp. 183-192.

70) Rosenfeld, Azriel, Picture Languages: Formal Languages for Picture Recognition, Academic Press, Inc., N.Y., 1979.

71) Rourke, J. O., and Badler, N. J., "Model Based Image Analysis of Human Motion Using Constrain Propagation," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 522-536.

72) Sashri, J. A., and Freeman, H., "Analysis of Generalized Chain Codes for the Representation of Planar Curves," IEEE Trans., Vol. PAMI-3, No. 5, September 1981, pp. 533-539.

73) Sankar, P. V., and Skalansky, J., A Gestalt-Guided Heuristic Boundary Follower for X-ray Images of Lung Nodules," IEEE Trans., Vol. PAMI-4, No. 3, May 1982, pp. 326-331.

74) Shackil, A. F., "An Electronic Human Eye," IEEE Spectrum, September 1980, pp. 88-91, Vol. 17, No. 9.

75) Shore, J. E., and Johnson, R. W., "Axomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross Entropy," IEEE Trans., Vol. IT-20, No. 1, January 1980.

76) Shore, J. E., "Minimum Cross Entropy Spectral Analysis," IEEE Trans., Vol. ASSP-29, No. 2, April 1981, pp. 230-237.

77) Shore, J. E., and Gray, R. M., "Minimum Cross Entropy Pattern Classification and Cluster Analysis," IEE Trans., Vol. PAMI-4, No. 1, January 1982, pp. 11-17.

78) Snyder, W. E., and Tang, D. E., "Findings the Extrema of a Region," IEEE Trans., Vol. PAMI-2, No. 3, 1981, pp. 261-269.

79) Stockman, G., Kopstein, S., and Benett, S., "Matching Images to Models for Registration and Object Detection via Clustering," IEEE Trans., Vol. PAMI-4, No. 3, May 1982, pp. 229-241.

80) Stuller, J. A., and Netravali, A. N., "Transform Domain Motion Estimation," The Bell System Technical Journal, Vol. 58, No. 7, September 1979, pp. 1673-1717.

81) Tanimoto, S., and Klinger, A., Structured Computer Vision, Academic Press, Inc., N.Y., 1980.

82) Tang, G. Y., and Huang, T. S., "Using the Creation Machine to Locate Airplanes on Aerial Photos," Pattern Recognition, Vol. 12, 1980, pp. 431-442, Pergamon Press, Ltd., N.Y.

83) Tejwani, Y. J., and Jones, R. A., "On the Detection of Peaks and Valleys Using the Local Descriptors Method," Submitted to IEEE Trans. PAMI.

84) Thompson, W. B., "Combining Motion & Contrast for Segementation," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 543-549.

85) Thomas, S. J., Louis, R. E., and Malindzak, G. S., "An Algorithm for Line Intersection Identification," Pattern Recognition, Vol. 13, No. 2, 1981, pp. 159-165.

86) Tsotso, J. K., Mylopoulous, J., Covvey, H. D., and Zucker, S. W., A Framework for Visual Motion Understanding," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 563-573.

87) Tsjui, S., Osada, M., and Yachida, M., "Tracking & Segmentation of Moving Objects in Dynamic Line Images," IEEE Trans. Pattern. Analy. Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 516-522.

88) Ullman, S., The Interpretation of Visual Motion, The MIT Press, MA, 1979.

89) Ulrych, T., "Maximum Entropy Power Spectrum of Long Period Geomagnetic Reversals," Nature, Vol. 235, January 28, 1972, pp. 218-219.

90) Valerie, Preston, Dunlop, "A Notation System for Recording Observable Motion," Int. J. Man-Machine Studies, 1969, pp. 361-386.

91) Wallace, T. P., Mitchel, O. R., and Fukunaga, K. "Three Dimensional Shape Analysis using Local Shape Descriptors," IEEE Trans., Vol. PAMI-3, No. 3, May 1981, pp. 310-323.

92) Wantanbe, S., "Pattern Recognition as a Quest for Minimum Entropy," Pattern Recognition, Vol. 13, No. 5, 1981, pp. 381-388.

93) Wintson, P. H., The Psychology of Computer Vision, McGraw-Hill, 1975.

94) Wintson, P. H., Artificial Intelligence, Addison-Wesley, Reading, M.A., 1977.

95) Yachida, M., Ikeda, M., and Tsuji, S., "A Plan Guided Analysis of Cineagrorams for Measurement of Dynamic Behavior of Heart Wall," IEEE Trans. Pattern. Analy. Machine Intelligence," Vol. PAMI-2, No. 6, November 1980, pp. 537-543.

96) Yachida, M., Asada, M., and Tsuji, S., "Automatic Analysis of Moving Images," IEEE Trans. Pattern. Analy. Machine Intelligence, PAMI, January 1981, pp. 12-20.

97) Yamada, Y., Tazaki, S., and Gray, R. M., "Asymptotic Performance of Block Quantizers with Difference Distortion Measures," IEEE Trans., Vol. IT-26, No. 1, January 1980, pp. 6-14.

98) Zahn, C. T., and Roskies, R. Z., "Fourier Descriptors for Plane Closed Curves," IEEE Trans., Vol. C-12, March 1972, pp. 269-281.

APPENDICES

APPENDIX A

DETAILS OF THE PROGRAM FOR THE ADAPTIVE LINE OF SIGHT METHOD.

Refering to the flow-chart shown in FIG. 27. Side A refers to the condition when the computations are performed from I to J modulo the number of points in the shape, while side B refers to the condition when computations are performed from J to to I. Forward track or Ftrack denotes to a condition when I is held constant while J is incremented. Backtrack or Btrack denotes a condition when J-1 is held constant while I is decremented.

Initially I and J are always chosen to be adjacent points, first going in the clockwise direction then in the anti-clockwise direction The details of the computational block, the detection block/process and some other blocks are as follows

COMPUTATIONAL BLOCK

Find the equation of the straight line L joining I to J

Find the distance DISTAIJ between points I and J.

Find the equations of the straight lines normal to L and joining every point P in between I and J.

Find the intersection (XINTSEIJ , YINTSEIJ) of each of

FLOWCHART FOR THE ADAPTIVE LINE OF SIGHT ALGORITHM

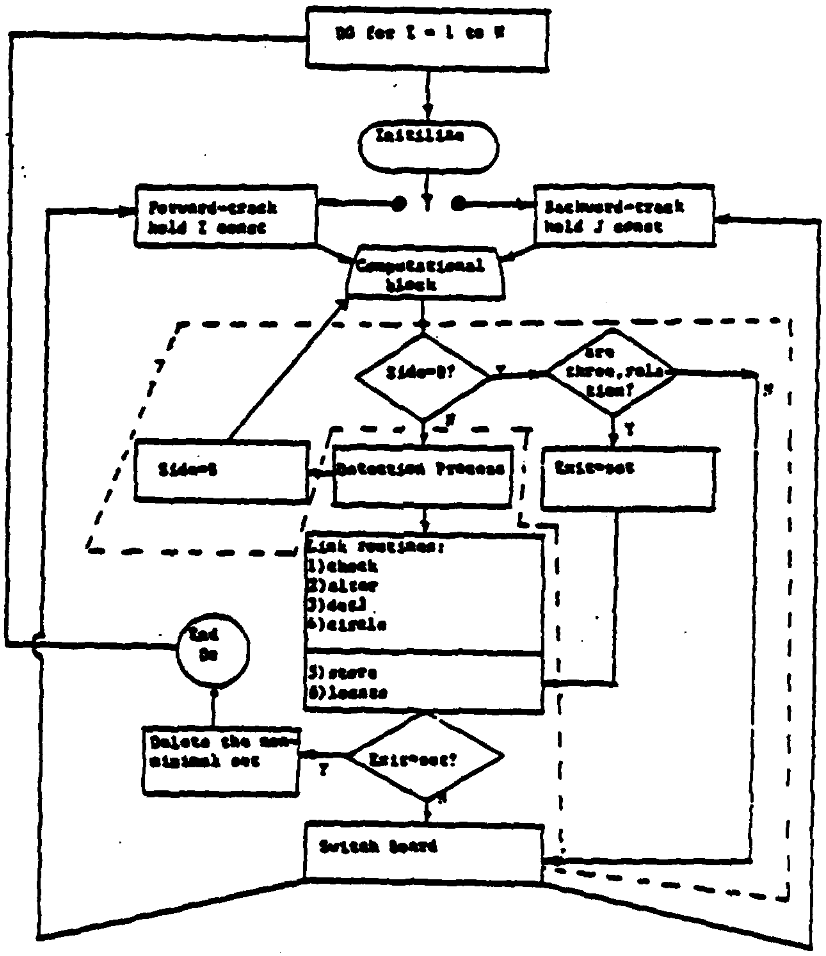


FIG. 27. FLOWCHART FOR THE ADAPTIVE LINE OF SIGHT METHOD.

the normal lines found in the above step with the straight line L

Find the distances DISTXIIJ and DISTYIIJ from point I to the intersection and from point J to the intersection, respectively for every P.

Find the normal distance NDISTAIJ from every P to the straight line L.

Find the normal vectors from the straight line to every point P.

DETECTION PROCESS

In this block a detect switch is set indicating that a critical point has been detected at I and J-1, if at the first instance, a point is found which is not on the same side of L as other points, or a point cannot be mapped injectively on to the straight line L. The former condition is checked by comparing the magnitude of the sum of every two adjacent normal vectors with the magnitude of those forming the sum $\pm$ a THRESHOLD1, while the latter condition is checked by comparing the sum of the distance DISTXIIJ+DISTYIIJ to DISTAIJ $\pm$ THRESHOLD2. Where thresholds 1 and 2 are set to account for round-off, truncation, quantization and other errors.

SWITCH BOARD

This is a control block which forces the computations to occur in an alternating sequence FORWARD-TRACK -BACK-TRACK -FORWARD- TRACK.......

ARE THE TWO RELATIONS?

In path B the two relations are not satisfied if one of the following is true.

1) The x or y components of the intersections, i.e., XINTSEIJ and YINTSEIJ respectively, are not monotonic.

2) The sum of the distances DISTXIIJ + DISTYIIJ is not equal to the distance DISTAIJ ± THRESHOLD2

APPENDIX B

DETAILS OF THE FEATURE VECTOR FORMATION PROGRAM.

To find whether a point within $(n \pm 3)$ of the nth critical point belongs to the nth feature it is not necessary to carry out the computation for each of the points within $(n \pm 3)$. Specifically the following two facts have been exploited to reduce computation:

1) Points within $(n \pm 1)$ are always in sight of C_n
2) The Line of Sight of a Point is a symmetric relation.

Let the relation (is in Line of Sight of the Point) be denoted by R; then if C_i R C_h implies C_h R C_i. Once it is determined that C_i R C_h then it is not necessary to compute if C_h R C_i. In context with, our example,

$$C_n \; R \; C_{n+2} \quad \text{implies } C_{n+2} \; R \; C_{n-2}$$

and

$$C_n \; R \; C_{n+3} \quad \text{implies } C_{n+3} \; R \; C_{n-3}$$

Thus it is necessary to check only if the critical point C_{n+2} and C_{n+3} are in Line of Sight of C_n.

If, the point of intersection of the line joining C_n to C_{n+k}, k=2, 3., with a line between C_j and C_{j+1} j=1...N., is denoted by INT, and the coordinate of the

following points are,

INT $=$ (XINT, YINT),

C_n $=$ (FEAT1X, FEAT1Y),

C_{n+k} $=$ (FEAT2X, FEAT2Y),

C_j $=$ (CURVE1X, CURVE1Y),

C_{j+1} $=$ (CURVE2X, CURVE2X),

and the distances between the following points are,

INT and C_n $=$ DIF1,

INT and C_{n+k} $=$ DIF2,

INT and C_j $=$ DIC1,

INT and C_{j+1} $=$ DIC2,

C_n and C_{n+k} $=$ DISFEA,

C_j and C_{j+1} $=$ CURVED,

then the relation R is not true if

$$\text{DIF1} + \text{DIF2} = \text{DISFEA}$$

or

$$\text{DIC1} + \text{DIC2} = \text{CURVED}.$$

The matrix SWSET in the program is this binary relation table. The output of the first proc matrix has seven columns. The first column corresponds to the sequence number of the critical point in the critical point list. The next two columns are the coordinates of the critical points. The last four columns contain the boolean relations operator. Specifically they hold the following the binary relations:

$C_{n-3}\ R\ C_n$, $C_{n-2}\ R\ C_n$, $C_{n+2}\ R\ C_n$, and $C_{n+3}\ R\ C_n$.

APPENDIX C

DETAILS OF THE PROGRAM FOR POST PROCESSING.

This program takes as input the critical points which are represented by their sequence number in the shape data list, and process them sequentially according to the following steps:

1) Every set of five consecutive points is replaced by their median point.

2) Every set of four consecutive points is replaced by the second point if FRC = SET otherwise by the third point if FLC = SET. Both FRC and FLC cannot be equal to SET at the same time.

3) Every set of three consecutive points is replaced by their median point.

4) Every set of two consecutive point is replaced by the first point if TLC = SET otherwise by the second point if TRC = SET. Both TLC and TRC cannot be equal to SET at the same time.

PROGRAM LISTING FOR THE ADAPTIVE LINE OF SIGHT METHOD.

INPUT: Three column data set named SHAPEXY. The first column represents the sequence number of data points in the shape data list. The 2nd and the 3rd columns are the x and y coordinates of the shape data points.

OUTPUT: 1) A 10 column matrix array named INF. The columns contain the following;

1st column = The number of critical point pair (the Ith and the Jth point) found in the first pass.

2nd column = sequence number of the Ith critical point in the shape data list.

3rd and 4th columns = the x and y coordinates of the Ith point in column 2.

5th column = sequence number of the Jth critical point in the shape data list.

6th and 7th columns = the x and y coordinates of the Jth critical point in column 5.

8th column = the location (sequence number in the shape data list) of a maxima or a minima in-between the Ith and the Jth point.

9th and 10th columns = the x and y coordinates of the point whose sequence number is in the 8th column.

OUTPUT 2) The input data.

```
*********START SYNCHRONISE;
PROC MATRIX ;
FETCH SHAPEXY;
NM=NROW(SHAPEXY);
RESET=0;
SET=1;
FTRACK=SET;
BTRACK=RESET;
DISCON=RESET;
CONNEC=SET;
PATHB=DISCON;
SIDEB=SET;
OPSIGHT=SET;
************* STARTING AND ENDING POINTS FOR THE ALGORITHM ARE I AND J;
I=1;
J=2;
ITEXP=I;
JTEXP=J;
NN=1;
************* NN IS A INDEX WHICH POINTS TO THE NEXT CRITICAL POINT;

************* NEED A MATRIX TO STORE INFORMATION ABOUT A CRITICAL POINT
say the expected number of critical points is 75 so dimension INF;
INF=J.(75,10,0);
*;
*;
*;
*;
**********************************************************************
**********************************************************************
**********************************************************************;
*COMPUTATION BLOCK;
START:

DETECT1=RESET;
DETECT2=RESET;
DETECT3=RESET;
DETECT=RESET;
EXIT=RESET;




AQ=SHAPEXY(J,3)-SHAPEXY(I,3);
BQ=SHAPEXY(J,2)-SHAPEXY(I,2);
CQ=SHAPEXY(J,2)#SHAPEXY(I,3);
DQ=SHAPEXY(I,2)#SHAPEXY(J,3);
SLOPEIJ=AQ#/BQ;
INTERIJ=(CQ-DQ)#/BQ;
DISTAIJ=SQRT((BQ#BQ)+(AQ#AQ));
```

```
***********************************************************************
*
*NEED TO KNOW HOW MANY POINTS ARE IN BETWEEN;
***********************************************************************
*;
IF (I>J)
THEN
DO;
PQ=NM-(I+1);
PT=J+PQ;
END;
ELSE
DO;
PT=J-(I+1);
END;
IF PT=0
THEN
DO;
GO TO SWBOARD;
END;
***********************************************************************
DIMENSION THE MATRICES FOR POINTS IN BETWEEN
***********************************************************************;



NINTERCY=J.(PT,1,0);
XINTSEIJ=J.(PT,1,0);
YINTSEIJ=J.(PT,1,0);
KDISTAIJ=J.(PT,1,0);
DISTXIIJ=J.(PT,1,0);
DISTYIIJ=J.(PT,1,0);
VECTORXI=J.(PT,1,0);
VECTORYI=J.(PT,1,0);
SUMVECXI=J.(PT,1,0);
SUMVECYJ=J.(PT,1,0);
MAGVECXY=J.(PT,1,0);




***********************************************************************
*
CERTAIN OPERATIONS NEED TO BE PERFORMED BETWEEN POINTS I AND J
***********************************************************************
;

CONSVEX=0;
CONSVEY=0;
```

```
DO K=1 TO PT BY 1;



II=K+I+NN;
II=MOD(II,NN);
IF (II=0) THEN DO; II=NN; END;
ASLOPEIJ=ABS(SLOPEIJ);
IF (ASLOPEIJ < 0.0000001)
THEN
DO;
XINTSEIJ(K,)=SHAPEXY(II,2);
YINTSEIJ(K,)=SHAPEXY(I,3);
END;
ELSE
DO;
NS=-1*/SLOPEIJ;
NINTERCY(K,)=SHAPEXY(II,3)-(SHAPEXY(II,2)*NS);
XINTSEIJ(K,)=(NINTERCY(K,)-INTERIJ)*/(SLOPEIJ-NS);
YINTSEIJ(K,)=(NS*XINTSEIJ(K,)+NINTERCY(K,));
END;
*;
*;
AQ=SHAPEXY(II,2)-XINTSEIJ(K,);
BQ=SHAPEXY(II,3)-YINTSEIJ(K,);
CQ=SHAPEXY(I,2)-XINTSEIJ(K,);
DQ=SHAPEXY(I,3)-YINTSEIJ(K,);
FQ=SHAPEXY(J,2)-XINTSEIJ(K,);
GQ=SHAPEXY(J,3)-YINTSEIJ(K,);
*;
*;
DISTXIIJ(K,)=SQRT((CQ*CQ)+(DQ*DQ));
DISTYIIJ(K,)=SQRT((FQ*FQ)+(GQ*GQ));
NDISTAIJ(K,)=SQRT(AQ*AQ+BQ*BQ);
VECTORXI(K,)=AQ;
VECTORYI(K,)=BQ;
SUMVECXI(K,)=CONSVEX+VECTORXI(K,);
SUMVECYJ(K,)=CONSVEY+VECTORYI(K,);
MAGVECXY(K,)=SQRT((SUMVECXI(K,)**2)+(SUMVECXI(K,)**2));
CONSVEX=VECTORXI(K,);
CONSVEY=VECTORYI(K,);
*;
*;
*;
*;
*;
*;
END;
*;
*;
*;
```

```
IF (PATHB=DISCON) THEN GO TO SIDEA;
IF (SIDEB=SET) THEN
DO;
LINK SIGET;
IF (OPSIGHT=RESET) THEN GO TO QUIT;
IF (OPSIGHT=SET) THEN
DO;
LINK LOCATEX;
EXIT=SET;
GO TO QUIT;
END;
END;
SIDEA:
**********************************************************************
**********************************************************************
ENTER DETECT PROCESSES
*********************************************************************;
************DIVERGE TO DETECT PROCESSES 1 AND 2 ********************;
********************************************************************;
*;
*;
*;
*;
*;
* PROCESS DETECT1;
DETECT1=RESET;
DO K=1 TO PT BY 1;
LL=DISTALJ-9.99;
UL=DISTALJ+9.99;
CL=DISTXILJ(K,)+DISTYILJ(K,);
IF ((CL < LL) OR (CL > UL))
THEN
DO;
DETECT1=SET;
GO TO CONVERG1;
END;
*;
*;
*;
END;
***********************************************************************
***
********************
*;
*;
*;
* DETECT PROCESS 2;
*;
CONNORXY=0.0;
*;
*;
DO K=1 TO PT BY 1;
AQ=NDISTALJ(K,)-29;
BQ=CONNORXY-29;
CONNORXY=NDISTALJ(K,);
```

```
IF ((MAGVECXY(K,) < AQ) OR (MAGVECXY(K,) < BQ))
THEN
DO;
DETECT2=SET;
GO TO CONVERG1;
END;
*;
*;
*;
END;
*********************************************************************
***;
*********************************************************************
***;
*;
*;
*;
*;
CONVERG1:
*;
*;
DETECT=RESET;
IF ((DETECT1 EQ SET) OR (DETECT2 EQ SET)) THEN DETECT=SET;
IF (DETECT NE SET)
THEN
DO;
GO TO SWBOARD;
END;
ELSE
DO;
IF (FTRACK=SET)
THEN
DO;
OLDPT=PT;
AZ=I;
BZ=J-1+NM;
BZ=MOD(BZ,NM);
IF (BZ=0) THEN BZ=NM;
OLDI=AZ;
LINK LOCATEI;
LINK LOCATEX;
LINK STOREI;
*LINK CHECK;
IF (EXIT=SET) THEN GO TO QUIT;
GO TO SWBOARD;
END;
IF (BTRACK=SET)
THEN
DO;
NEWPT=PT;
AZ=I+1+NM;
AZ=MOD(AZ,NM);
IF (AZ=0) THEN AZ=NM;
NEWI=AZ;
BJ=J;
IF (OLDI EQ NEWI) THEN GO TO SWBOARD;
```

```
LINK LOCATEI;
LINK LOCATEX;
LINK STOREI;
IF (EXIT=SET) THEN GO TO QUIT;
LINK DET3;
IF DETECT3=SET
THEN
DO;
LINK CIRCLE;
GO TO SWBOARD;
END;
IF DETECT3=RESET
THEN
DO;
LINK ALTER1;
GO TO SWBOARD;
END;
END;
*;
END;
*;
*;
FREE
 VECTORXI
 VECTORYI
 NINTERCY
 XINTSELJ
 YINTSELJ
 NDISTALJ
 DISTXILJ
 DISTYILJ
 SUMVECXI
 SUMVECYJ
 MAGVECXY;
*;
*;
*;
*;
*;
****************************************************************
;
*                               SWITCH BOARDS;
****************************************************************
;
SWBOARD:
*;
*;
*;
IF ((DETECT EQ RESET) & (FTRACK EQ SET))
THEN
DO;
J=J+1+NN;
J=MOD(J,NN);
IF (J=0) THEN J=NN;
GO TO START;
END;
```

```
*;
*;
*;
IF ((DETECT=RESET) & (BTRACK=SET))
THEN
DO;
I=I-1+NN;
I=MOD(I,NN);
IF (I=0) THEN I=NN;
GO TO START;
END;
*;
*;
*;
IF ((DETECT EQ SET) & (FTRACK EQ SET))
THEN
DO;
FBTRACK=SET;
FTRACK=RESET;
BTRACK=SET;
J=J-1+NN;
J=MOD(J,NN);
IF (J=0) THEN J=NN;
I=I-1+NN;
I=MOD(I,NN);
IF (I=0) THEN I=NN;
GO TO START;
END;
IF ((DETECT=SET) & (BTRACK=SET))
THEN
DO;
BFTRACK=SET;
FTRACK=SET;
BTRACK=RESET;
I=J+NN;
I=MOD(I,NN);
IF (I=0) THEN I=NN;
J=I+1+NN;
J=MOD(J,NN);
IF (J=0) THEN J=NN;
IF (PATHB=DISCON) THEN GO TO STRAIGHT;
IF (SIDEB=RESET) THEN
DO;
SIDEB=SET;
ITEXP=I;
JTEXP=J;
I=JTEXP;
J=ITEXP;
END;
GO TO START;
GETOUT:
SIDEB=RESET;
I=ITEXP;
J=JTEXP;
```

```
STRAIGHT:
GO TO START;
END;
**********************************************************************
;
*                     LINK ROUTINES
;
**********************************************************************
;
LOCATEI:
DO;
A=NDISTALJ(1,);
LOC=1;
NQ=PT-1;
DO III=2 TO NQ BY 1;
IF (A < NDISTALJ(III,) )
THEN
DO;
A=NDISTALJ(III,);
LOC=III;
END;
END;
END;
RETURN;
*;
*;
*;
LOCATEX:
IF ((PT<4) OR (NN>70)) THEN GO TO JUMPS;
*AZ=I+1+NN;
*AZ=MOD(AZ,NN);
*IF (AZ=0) THEN AZ=NN;
*BZ=J-1+NN;*BZ=MOD(BZ,NN);*IF (BZ=0) THEN BZ=NN;
NQ=PT-1;NQQ=PT-2;
NAN=NDISTALJ(1:NQ,);
NBN=NDISTALJ(2:PT,);
DABD=NAN-NBN;
DBAD=DABD(2:NQ,);
DBAD=DBAD#DABD(1:NQQ,);
DABSD=ABS(DBAD);
COUNTME=0;
DO IXXXI=1 TO NQQ;
AKME=DBAD(IXXXI,);
SKME=DABSD(IXXXI,);
IF ((AKME<0) AND (SKME>4)) THEN
DO;
INP(NN,1)=NN;
INP(NN,2)=AZ;
INP(NN,3:4)=SHAPEXY(AZ,2:3);
INP(NN,5)=BZ;
INP(NN,6:7)=SHAPEXY(BZ,2:3);
COUNTME=COUNTME+1;
MYLOC=IXXXI+NN+AZ+1;
MYLOC=MOD(MYLOC,NN);
IF (MYLOC=0) THEN MYLOC=NN;
*IF ((AKME < 0) AND (SKME > .27)) THEN DO;
INP(NN,8)=MYLOC;
```

```
INF(NN,9:10)=SHAPEXY(NYLOC,2:3);
IF (NN>70)
THEN
DO;
EXIT=SET;
GO TO JUMPS;
END;
NN=NN+1;
END;
END;
JUMPS:
IF (NN>70) THEN EXIT=SET;
RETURN;
*;
*;
*;
*;
STOREI:
INF(NN,1)=NN;
INF(NN,2)=AZ;
INF(NN,3:4)=SHAPEXY(AZ,2:3);
INF(NN,5)=BZ;
INF(NN,6:7)=SHAPEXY(BZ,2:3);
LOC=LOC+NN+AZ;
LOC=MOD(LOC,NN);
IF (LOC =0) THEN LOC=NN;
INF(NN,8)=LOC;
INF(NN,9:10)=SHAPEXY(LOC,2:3);
IF (NN > 70 )
THEN
DO;
EXIT=SET;
END;
NN=NN+1;
RETURN;
*;
*;
*;
CIRCLE:
RETURN;
*;
*;
*;
CHECK:
NY=NN-4;
DO KI= 1 TO NY;
AK=INF(KI,5);
BBBZ=BZ+2;
BBBY=BZ-2;
IF (BBBY < AK < BBBZ)
THEN
DO;
EXIT=RESET;
END;
```

```
END;
RETURN;
*;
*;
*;
SIGHT:
OPSIGHT=SET;
*if you assume zero threshold then use the go below
other wise delete the go to statement.;
GO TO INS;
OPSIGHT=RESET;
IF ((PT>4) OR (MN>70)) THEN GO TO JUMPM;
PRINT DISTAIJ;
DO YK=1 TO PT BY 1;
LL=DISTAIJ-.1;
UL=DISTAIJ+.1;
CL=DISTXIIJ(YK,) + DISTYIIJ(YK,);
PRINT UL LL CL;
IF ((CL<LL) OR (CL>UL))
THEN
DO;
NOTHING=SET;
END;
ELSE;
DO;
OPSIGHT=SET;
GO TO JUMPM;
END;
END;
INS:
NQ=PT-1;NQQ=PT-2;
Y1AY=XINTSEIJ(1:NQ,); Y2AY=XINTSEIJ(2:PT,);
YAY=Y1AY-Y2AY;
YA1Y=YAY(1:NQQ,);YA2Y=YAY(2:NQ,);
YAAY=YA1Y#YA2Y;
Y1BY=YINTSEIJ(1:NQ,);Y2BY=YINTSEIJ(2:PT,);
YBY=Y1BY-Y2BY;
YB1Y=YBY(1:NQQ,);YB2Y=YBY(2:NQ,);
YBBY=YB1Y#YB2Y;
PRINT YBBY YAAY ;
DO KME=1 TO NQQ;
YKYOU=YAAY(KME,);YLYOU=YBBY(KME,);
IF ((YKYOU < .000000) OR (YLYOU < .000000)) THEN
DO;
OPSIGHT=RESET;
END;
END;
JUMPM:
RETURN;
*;
*;
*;
*;
*;
```

```
ALTER1:
M1=NN-1;
M2=NN-2;
A=INF(M1,5);
B=INF(M2,5);
TEMP1=INF(M1,);
IF (A=B)
THEN
DO;
INF(M2,)=TEMP1;
NN=M1;
END;
RETURN;
*;
*;
*;
DET3:
IZZI=INF(NN,2);
IZZJ=INF(NN,5);
NUMM=NN-1;
IF (IZZI < IZZJ)
THEN
DO ISS=1 TO NUMM BY 1;
IF  ((IZZI <INF(IZZ,2) <IZZJ) OR (IZZJ <INF(IZZ,5) <IZZJ))
THEN
DO;
DETECT3=SET;
UNOTE=IZZ;
PRINT DETECT3 UNOTE;
END;
*;
END;
*;
IF (IZZI >IZZJ)
THEN
DO IZZ=1 TO NUMM BY 1;
INDEXI=INF(IZZ,2);
INDEXJ=INF(IZZ,5);
IF (((IZZI < INDEXI) OR (INDEXI <IZZJ)) OR
((IZZI < INDEXJ) OR (IZZI < INDEXJ)))
THEN
DO;
DETECT3=SET;
UNOTE=IZZ;
PRINT DETECT3 IZZ;
END;
END;
RETURN;
QUIT;
OUTPUT SHAPEXY OUT=SHAPEXY(RENAME=(COL1=J COL2=X COL3=Y));
OUTPUT INF OUT=INF(RENAME=(COL1=J COL2=II COL3=XII COL4=YII
 COL5=JJ COL6=XJJ COL7=YJJ COL8=LOC COL9=XLOC COL10=YLOC));
*;
```

FEATURE VECTOR FORMATION PROGRAM

INPUT: The input to the program is data set called SHAPEXZ. SHAPEXZ contains the X and Y coordinates of the critical pc nts.

OUTPUT: The output of the the program is a matrix called DICTION. DICTION has seven columns. The 1st column is the sequence number F_n to which the critical point belongs. The 2nd column is the sequence number of the critical point in the list of critical points. The 3rd column represent the binary relationship C_n R $C_{n\pm k}$, k=1, 2, 3,. The 4th and the 5th represent the x and y coordinates of the critical point if R = 1; The 6th and the 7th column contain the feature vector components s_{nk} and α_{nk} respectively.

```
PROC MATRIX ;FETCH SHAPEXZ;NR=NROW(SHAPEXZ);
NN=NR+6;
CON=J.(NN,7,.);
NRM1=NR-1;
NRP1=NR+1;
NNM1=NN-1;
CON(1:NR,1:2)=SHAPEXZ;
CON(NRP1:NN,1:2)=SHAPEXZ(1:6,);
CON(1:NNM1,3:4)=CON(2:NN,1:2);
CON(NN,3:4)=CON(7,1:2);
CON(1:NN,5:6)=CON(1:NN,1:2)-CON(1:NN,3:4);
CON(1:NN,7)=CON(1:NN,5)#CON(1:NN,5)+CON(1:NN,6)#CON(1:NN,6);
ABME=CON(1:NN,7);
CON(1:NN,7)=SQRT(ABME);
*;
*;
*;
DO HA =1 TO NN;
ABCD=CON(HA,5);
ABCD=ABS(ABCD);
IF (ABCD < .0000001) THEN
DO;
CON(HA,5)=9999999;
CON(HA,6)=000000;
END;
IF (ABCD > .0000001) THEN
DO;
CON(HA,5)=CON(HA,6)#/CON(HA,5);
CON( HA,6)=CON( HA,2)-CON( HA,5)#CON( HA,1);
END;
END;
*;
*;
*;
*;
*;
NNM3=NN-3;
NNM2=NN-2;
NNM5=NN-5;
SWSET=J.(NN,4,.);
DO I=1 TO NN;
IP1=I+1;
IP2=I+2;
IP3=I+3;
*;
*;
COUNT=1;
RENTER:
IF (COUNT=1) THEN IP=IP2;
IF (COUNT=2) THEN IP=IP3;
*;
IF ( IP > NN) THEN GO TO EXIT;
*;
FEAT1X=CON(I,1);FEAT1Y=CON(I,2);
```

```
FEAT2X=CON(IP,1);FEAT2Y=CON(IP,2);
*;
*;
F=CON(I,);G=CON(IP,);
*PRINT F G;
*EQUATION;
DELFEX=FEAT1X-FEAT2X;DELFEY=FEAT1Y-FEAT2Y;
DISFEA=DELFEX*DELFEX + DELFEY*DELFEY;
DISFEA=SQRT(DISFEA);
DTEST=ABS(DELFEX);
IF (DTEST =< .0000001) THEN
DO;
FM=99999999;
FC=0;
END;
IF (DTEST > .0000001) THEN
DO;
FM=DELFEY*/DELFEX;
FC=FEAT1Y-FM*FEAT1X;
END;
*;
*;
*;
*;
SIGHT=1;
*SIGHT;
DO NI= 1 TO NN;
CURVEM=CON(NI,5);
CURVEC=CON(NI,6);
CURVED=CON(NI,7);
DIFSLO=(FM-CURVEM);
DIFSLOP=ABS(DIFSLO);
IF (DIFSLOP > .0000001) THEN
DO;
IF (CURVEM >= 9999999) THEN
DO;
XINT=CON(NI,1);
YINT=FM*XINT+FC;
GO TO STEPX;
END;
*;
*;
IF (FM >= 9999999) THEN
DO;
XINT=FEAT1X;
YINT=CURVEM*XINT+CURVEC;
GO TO STEPX;
END;
XINT=(CURVEC-FC)*/DIFSLO;
YINT=FM*XINT+FC;
STEPX:
CURVE1X=CON(NI,1);CURVE1Y=CON(NI,2);
CURVE2X=CON(NI,3);CURVE2Y=CON(NI,4);
FSTXD1=FEAT1X-CURVE1X;
```

```
FSTYD1=FEAT1Y-CURVE1Y;
FSTXD2=FEAT2X-CURVE1X;
FSTYD2=FEAT2Y-CURVE1Y;
SECXD1=FEAT1X-CURVE2X;
SECYD1=FEAT1Y-CURVE2Y;
SECXD2=FEAT2X-CURVE2X;
SECYD2=FEAT2Y-CURVE2Y;
*;
*;
*;
IF (((FSTXD1=0) & (FSTYD1=0))OR((FSTXD2=0)&(FSTYD2=0))OR
((SECXD1=0)&(SECYD1=0)) OR ((SECXD2=0)&(SECYD2=0))) THEN GO TO OUT0;
DIF1X=FEAT1X-XINT;
DIF1Y=FEAT1Y-YINT;
DIF1=DIF1X*DIF1X+DIF1Y*DIF1Y;
DIF1=SQRT(DIF1);
DIF2X=FEAT2X-XINT;
DIF2Y=FEAT2Y-YINT;
DIF2=DIF2X*DIF2X+DIF2Y*DIF2Y;
DIF2=SQRT(DIF2);
DIC1X=CURVE1X-XINT;
DIC1Y=CURVE1Y-YINT;
DIC1=DIC1X*DIC1X+DIC1Y*DIC1Y;
DIC1=SQRT(DIC1);
DIC2X=CURVE2X-XINT;
DIC2Y=CURVE2Y-YINT;
DIC2=DIC2X*DIC2X+DIC2Y*DIC2Y;
DIC2=SQRT(DIC2);
SUMDIC=DIC1+DIC2;SUMDIF=DIF1+DIF2;
DIFF1=CURVED-SUMDIC;DIFF1=ABS(DIFF1);
DIFF2=DISFEA-SUMDIF; DIFF2=ABS(DIFF2);
*;
O=CON(NI,);
*PRINT O XINT YINT DIF1 DIF2 DIC1 DIC2 DISFEA;
*;
*;
*PRINT O;
*;
*;
*;
*;
IF ((DIFF1 < .000001) & (DIFF2 < .000001)) THEN
DO;
SIGHT=0;
APTEST=CON(I,1:2);
CPTEST=CON(IP,1:2);
BPTEST=CON(NI,);
*PRINT APTEST BPTEST CPTEST;
GO TO OUT1;
END;
END;
OUT0:
END;
OUT1:
```

```
*IF (((I=1)OR(I=3)OR(I=4)OR(I=31)OR(I=32)OR(I=33)OR(I=34)OR(I=38)))
*THEN
*DO;
*PRINT APTEST BPTEST CPTEST O XINT YINT DIF1 DIF2 DIC1 DIC2 DISFEA;
*END;
*PRINT SIGHT;
*;
*;
*;
*;
IF (COUNT=1) THEN
DO;
SWSET(I,3)=SIGHT;
COUNT=COUNT+1;
IF (I =< NNM3) THEN SWSET(IP2,2)=SIGHT;
GO TO RENTER;
END;
*;
*;
IF (COUNT=2) THEN
DO;
SWSET(I,4)=SIGHT;
IF (I =< NNM3) THEN SWSET(IP3,1)=SIGHT;
GO TO LOOPOUT;
END;
*;
*;
LOOPOUT:
END;
*;
EXIT:
*PRINT SWSET;
*;
CON(4:NNM3,3:6)=SWSET(4:NNM3,);
CON(1:3,3:6)=SWSET(NNM5:NNM3,);
CON(NNM2:NN,3:6)=SWSET(4:6,);
FREE SWSET;
*PRINT CON;
*;
*;
*;
OUTPUT CON OUT=CON(RENAME=(COL1=I1 COL2=I2 COL3=I3 COL4=I4 COL5=I5
COL6=I6 COL7=I7));
PROC PRINT DATA=CON;
DATA CON1;SET CON;J=_N_;DROP I7;
PROC DELETE DATA=CON;
PROC PRINT DATA=CON1;
*;
*;
*;
*;
PROC MATRIX; FETCH CON1; NN=NROW(CON1);
NN7=7#NN;
NNM5=NN-5;
```

```
CON1(NNN5:NN,7)=1/2/3/4/5/6;
NNM=NN7-42;
DICTION=J.(NNM,7,.);
NNN1=NN-1;
DO IW= 1 TO NNN1;
IWP1=IW+1;
IWP2=IW+2;
IWP3=IW+3;
IWP4=IW+4;
IWP5=IW+5;
IWP6=IW+6;
IWP7=IW+7;
IW7=IW#7;
IW7N6=IW7-6;
IW7N5=IW7-5;
IW7N4=IW7-4;
IW7N3=IW7-3;
IW7N2=IW7-2;
IW7N1=IW7-1;
IF ((IWP6 > NN) OR (IWP4 > NN) ) THEN GO TO RACK;
ANE=CON1(IWP3,3:4);ANF=CON1(IWP3,5:6);
DICTION(IW7N6:IW7,4:5)=CON1(IW:IWP6,1:2);
DICTION(IW7N6:IW7,2)=CON1(IW:IWP6,7);
DICTION(IW7N6:IW7,1)=IWP3#J.(7,1,1);
DICTION(IW7N4:IW7N2,3)=J.(3,1,1);
DICTION(IW7N6:IW7N5,3)=ANE';
DICTION(IW7N1:IW7,3)=ANF';
END;
RACK:
FREE CON1;
*;
*;
*;
DO I11=1 TO NNM;
AFDY=DICTION(I11,3);
IF (AFDY=0) THEN DICTION(I11,4:5)=J.(1,2,0);
END;
PRINT DICTION;
*;
*;
*;
;
*;
* SHAPE VECTOR CALCULATIONS;
*;
*;;
*;
*;
DO I33I=TO NN;
I33I7=I33
*;
*;
*;
*;
```

```
DO I33I=1 TO NN;
I33I7=7#I33I;I33IN6=I33I7-6;
D=J.(7,7,.);XI=J.(7,1,.);YI=J.(7,1,.);SD=J.(7,1,.);DI=J.(7,1,.);PHY=J
,1,.);
D=DICTION(I33IN6:I33I7,);XI=D(,4);YI=D(,5);NU=D(,3);
SIGMAX=XI(+,);SIGMAY=YI(+,);N=NU(+,);
CENTROIX=SIGMAX#/N;
CENTROIY=SIGMAY#/N;
DI=XI#XI+YI#YI;
DICTION(I33IN6:I33I7,6)=DI;
CANGLE=CENTROIX#/CENTROIY;
IF (CENTROIX > 0) THEN REF=ARCTAN(CANGLE);
IF (CENTROIX < 0) THEN REF=3.14+ARCTAN(CANGLE);
IF (( CENTROIX=0) AND (CENTROIY > 0)) THEN REF=1.57;
IF ((CENTROIX=0) AND (CENTROIY < 0) ) THEN REF=4.71;
*;
*;
*;
*;
DO INU=1 TO 7;
IN=I33IN6+INU-1;
IF (DICTION(IN,3)=0) THEN
DO;
DICTION(IN,6)=0;
IX(INU,)=0;
IY(INU,)=0;
DICTION(IN,7)=.;
END;
IF (DICTION(IN,3) NE 0) THEN
DO;
PHY(INU,)=IY(INU,)#/IX(INU,);
A=PHY(INU,);
*;
*;
IF (IX(INU,) > 0) THEN DICTION(IN,7)=ARCTAN(A);
IF (IX(INU,) < 0) THEN DICTION(IN,7)=3.14+ARCTAN(A);
IF ((IX(INU,)=0) AND (IY(INU,) > 0)) THEN DICTION(IN,7)=1.57;
IF ((IX(INU,)=0) AND (IY(INU,) < 0)) THEN DICTION(IN,7)=4.71;
DICTION(IN,7)=REF-DICTION(IN,7);
AGDIF=DICTION(IN,7);
AGDIF=COS(AGDIF);
AGDIF=ARCOS(AGDIF);
DICTION(IN,7)=AGDIF;
END;
END;
PRINT D;
END;
```

PROGRAM LISTING FOR THE POST PROCESSING STEP

INPUT: A data array named AINF which consists of one column of numbers representing the sequence number of critical points in the SHAPE DATA LIST.

OUTPUT: The output is a post processed data array AINF.

For more details see Appendix C

```
PROC MATRIX;FETCH AINF;NAINF=NROW(AINF);
RESET=0;SET=1;
FRC=SET;FLC=RESET;
TLC=RESET;TRC=SET;
WEYCOUNT=0;
NAINF3=NAINF-3;NAINF2=NAINF-2;NAINF1=NAINF-1;
NAINF4=NAINF-4;
*;
*;
DO MYW=1 TO NAINF4;
MVW=MYW+4;
AKMESTO5=AINF(MYW:MVW,);
AYOU1=AKMESTO5(1,);AYOU2=AKMESTO5(2,);AYOU3=AKMESTO5(3,);
AYOU4=AKMESTO5(4,);AYOU5=AKMESTO5(5,);
DYU1=AYOU1-AYOU2;DYU1=ABS(DYU1);
DYU2=AYOU2-AYOU3;DYU2=ABS(DYU2);DYU3=AYOU3-AYOU4;DYU3=ABS(DYU3);
DYU4=AYOU4-AYOU5;DYU4=ABS(DYU4);
IF ((DYU1=1) AND (DYU2=1) AND (DYU3=1) AND (DYU4=1)) THEN
DO;
MYWP5=MYW+5;
AINF(MYW,)=AYOU3;
AINF(MYWP1:NAINF4,)=AINF(MYWP5:NAINF,);
AINF(NAINF3:NAINF,)=J.(4,1,.);
END;
END;
*;
*;
DO MYW=1 TO NAINF3;
MVW=MYW+3;
AKMESTO4=AINF(MYW:MVW,);
AXXX1=AKMESTO4(1,);
AXXX2=AKMESTO4(2,);
AXXX3=AKMESTO4(3,);
AXXX4=AKMESTO4(4,);
ADXXX1=AXXX1-AXXX2;ADXXX2=AXXX2-AXXX3;ADXXX3=AXXX3-AXXX4;
ADXXX1=ABS(ADXXX1);ADXXX2=ABS(ADXXX2);ADXXX3=ABS(ADXXX3);
IF ((ADXXX1 EQ 1) AND (ADXXX3 EQ 1) AND (ADXXX2 EQ 1)) THEN
DO;
MYWP1=MYW+1;MYWP2=MYW+2;MYWP3=MYW+3;MYWP4=MYW+4;
IF (FLC=SET) THEN
DO;
AINF(MYW,)=AXXX2;
END;
IF (FRC=SET) THEN
DO;
AINF(MYW,)=AXXX3;
END;
AINF(MYWP1:NAINF3,)=AINF(MYWP4:NAINF,);
AINF(NAINF2:NAINF,)=J.(3,1,.);
END;
*;
*;
END;
*;
```

```
DO MYW=1 TO NAINF2;
MVW=MYW+2;
AKMESTO3=AINF(MYW:MVW,);
AD1=AKMESTO3(1,);
AD3=AKMESTO3(3,);
AD13DA=AD1-AD3;AD13DA=ABS(AD13DA);
IF (AD13DA EQ 2) THEN
DO;MYWP1=MYW+1;MYWP2=MYW+2;MYWP3=MYW+3;
MONKEY=AD1+1;
AINF(MYW,)=MONKEY;
AINF(MYWP1:NAINF2,)=AINF(MYWP3:NAINF,);
AINF(NAINF1:NAINF,)=J.(2,1,.);
GO TO BOTTOM3;
END;
*;
*;
*;
MYWP1=MYW+1;MYWP2=MYW+2;
AD2=AKMESTO3(2,);
AD12DA=AD1-AD2;
AD12DA=ABS(AD12DA);
IF (AD12DA EQ 2 ) THEN
DO;
MONKEY=AD1+1;
AINF(MYW,)=MONKEY;
AINF(MYWP1:NAINF1,)=AINF(MYWP2:NAINF,);
AINF(NAINF,)=J.(1,1,.);
END;
*;
*;
BOTTOM3:
END;
*;
*;
*;
DO MYW=1 TO NAINF2;
MVY=MYW+1;
AKMESTO2=AINF(MYW:MVY,);
ADDY=AKMESTO2(1,)-AKMESTO2(2,);
ADDY=ABS(ADDY);
IF (ADDY EQ 1) THEN
DO;
IF (TRC=SET) THEN
DO;
AINF(MYW,)=AKMESTO2(2,);
END;
MYWP1=MYW+1;MYWP2=MYW+2;
AINF(MYWP1:NAINF1,)=AINF(MYWP2:NAINF,);
AINF(NAINF,)=J.(1,1,.);
END;
END;
*;
*;
OUTPUT AINF OUT=AINF(RENAME=(COL1=JJ));
```

PROGRAM LISTING OF THE REVISED FEATURE VECTOR.

INPUT: Data set named AAUU with three columns. The 1st column represents the sequence number of the critical points in the critical point list, while columns 2 and 3 represent the x and y coordinates of the critical points.

OUTPUT: The program has two output matrices, 1) STOREE, 2) SKNO

STOREE has six columns which contain the following:

1st = feature number F_n to which the critical point in the second column belongs.

2nd = sequence number of the critical point in the critical point list.

3rd and 4th = the x and y coordiantes of the critical point.

5th and the 6th = the distance and the angle component of the feature vector, namely s_{nk} and α_{nk}.

SKNO has six columns which contain the following:

1st = x coordinate of the mean of the feature.

2nd = y coordinate of the mean of the feature.

3rd = the distance of the centroid of the feature from a

fixed reference.

4th = angle made by the centroid with a reference direction.

5th = size.

6th = feature number F_n.

```
PROC MATRIX;FETCH AAUU;NX=NROW(AAUU);NM3=3#NX-6;STOREE=J.(NM3,6,.);
AAUU=50#AAUU;
NO=NX-2;
*;
*;
DO I=1 TO NO;
IP1=I+1;IF (IP1 > NO ) THEN IP1=IP1-NO;
IM3=3#I;
IM3M2=IM3-2;
IP2=I+2;
STOREE(IM3M2:IM3,1)=IP1#J.(3,1,1);
STOREE(IM3M2:IM3,2:4)=AAUU(I:IP2,1:3);
XK=STOREE(IM3M2:IM3,3);
YK=STOREE(IM3M2:IM3,4);
SUMXK=XK(+,);
SUMYK=YK(+,);
MEANX=SUMXK#/3;
MEANY=SUMYK#/3;
CENT=MEANX#MEANX + MEANY#MEANY;
CENTRO=SQRT(CENT);
XK=XK-MEANX#J.(3,1,1);
YK=YK-MEANY#J.(3,1,1);
DK=XK#XK + YK#YK;
DK=SQRT(DK);
SIZE=DK(+,);
STOREE(IM3M2:IM3,5)=DK#/SIZE;
*;
*;
IF ( (MEANX = 0) AND (MEANY > 0)) THEN DO;
CENANGLE=1.57;
END;
IF ( (MEANX =0) AND (MEANY < 0)) THEN DO;
CENANGLE=4.17;
END;
IF (MEANX > 0) THEN DO;
CANG=MEANY#/MEANX;
CANG=ATAN(CANG);
CENANGLE=CANG;
END;
IF (MEANX < 0) THEN DO;
CANG=MEANY#/MEANX;
CANG=ATAN(CANG);
CENANGLE=CANG+3.14;
END;
*;
*;
DO IA=1 TO 3;
IF ((XK(IA,)=0) AND (YK(IA,) >0))
THEN DO;
PHI=3.14#/2;IQ=IM3M2+IA-1;
STOREE(IQ,6)=PHI;
END;
*;
IF ((XK(IA,)=0) AND (YK(IA,) < 0))
```

```
THEN DO;
IQ=IM3M2+IA-1;
PHI=4.71;
STOREE(IQ,6)=PHI;
END;
*;
IF (XX(IA,) > 0) THEN DO;
PHI=YK(IA,)#/XK(IA,);
PHI=ATAN(PHI);
IQ=IM3M2+IA-1;
STOREE(IQ,6)=PHI;
END;
*;
IF (XX(IA ,) < 0) THEN DO;
PHI=YK(IA,)#/XK(IA,);
PHI=ATAN(PHI);
PHI=3.14 + PHI;
IQ=IM3M2+IA-1;
STOREE(IQ,6)=PHI;
END;
*;
END;
*;
SKNO=J.(1,6,.);
SKNO(1,6)=IP1;
SKNO(1,1)=MEANX;
SKNO(1,2)=MEANY;
SKNO(1,3)=CENTRO;
SKNO(1,4)=CENANGLE;
SKNO(1,5)=SIZE;
PRINT SKNO;
*PRINT MEANX MEANY CENTRO CENANGLE SIZE;
*;
*;
END;
PRINT STOREE;
```

PROGRAM LISTING FOR THE MATCHING PROCESS.

INPUT: An eight column data set called GETI. The first four columns correspond to the data from the dictionary while the next four to the data from the problem text. The first of the four columns corresponds to the sequence number of the feature F_n, the 2nd corresponds to the ofsight code of the feature, and the 3rd and the 4th corresponds to the distance and angle components of the feature vector.

A word is picked and compared with features in the dictionary with the same ofsight code.

OUTPUT: Plots for RVALUE and SUMABSDIF of the word against features in the dictionary with the same ofsight code.

```
PROC MATRIX;FETCH GETI; NX=NROW(GETI);
WORD=GETI( 127:133,5:8);
*PRINT WORD;
*;
*;
*;
*;
O=J.(1,3,0);
STORE=J.(1,3,.);
DO I=1 TO NX BY 7;
I=I;I2=I+1;I3=I+2;I4=I+3;I5=I+4;I6=I+5;I7=I+6;
OFS1=GETI(I,2)-WORD(1,2);
OFS2=GETI(I2,2)-WORD(2,2);
OFS3=GETI(I3,2)-WORD(3,2);
OFS4=GETI(I4,2)-WORD(4,2);
OFS5=GETI(I5,2)-WORD(5,2);
OFS6=GETI(I6,2)-WORD(6,2);
OFS7=GETI(I7,2)-WORD(7,2);
*;
*;
*;
IF ((OFS1 NE 0) OR (OFS2 NE 0) OR (OFS3 NE 0) OR (OFS4 NE 0)
OR (OFS5 NE 0) OR (OFS6 NE 0) OR (OFS6 NE 0)) THEN
GO TO NEXT;
*;
*;
*;
METRIC =GETI(I:I7,3)-WORD( ,3);
*PRINT METRIC;
METRIC=ABS(METRIC);
*PRINT METRIC;
DIF=METRIC(+,);
ANGDIF=GETI(I:I7,4)-WORD( ,4);
COSDEL=COS(ANGDIF);SINDEL=SIN(ANGDIF);
*;
IF ( I = 120) THEN DO;
*PRINT  ANGDIF COSDEL SINDEL;END;
*;
SUMDIFC=COSDEL(+,);SUMDIFS=SINDEL(+,);
SQAC=SUMDIFC#SUMDIFC+SUMDIFS#SUMDIFS;
IF (I=120) THEN DO;
*PRINT SUMDIFC SUMDIFS SQAC;END;
SS=SQRT(SQAC);
NN=WORD( ,2);
N=NN(+,);
RVALUE=SS#/N;
IF (I=120) THEN DO;
*PRINT SS N;END;
STORE(1,1)=GETI(I4,1);
STORE(1,2)=DIF;
STORE(1,3)=RVALUE;
*;
*;
*;
```

```
O=O//STORE;
NEXT;
END;
O(1,)=J.(1,3,.);
OUTPUT O OUT=O (RENAME=(COL1=FEATURE COL2=ABSSUM COL3=RVALUE));
*;
*;
*;
*PROC PRINT DATA=O;
PROC GPLOT DATA=O;
*PLOT ABSSUM*FEATURE=1 RVALUE*FEATURE=2 / OVERLAY;
*PLOT ABSSUM*FEATURE=1;
PLOT ABSSUM*FEATURE=1/ HZERO VZERO;
*PLOT ABSSUM*RVALUE=FEATURE /I=M3X;
SYMBOL1 I=NEEDLE V=DIAMOND;
TITLE ABSOLUTE DIFFERENCE AGAINST FEATURE NO.;
FOOTNOTE1 ONLY FEATURES OF THE SAME OFSIGHT INFORMATION ARE COMPARED;
PROC GPLOT;
*PLOT RVALUE*FEATURE;
PLOT RVALUE*FEATURE=1/ HZERO VZERO;
TITLE ANGULAR CORRELATION AGAINST FEATURE NO.;
FOOTNOTE1 ONLY FEATURES OF THE SAME OFSIGHT INFORMATION ARE COMPARED;
```

LISTING OF THE PROGRAM WHICH WAS USED TO PLOT
FIG. 17 'EXAMPLES OF SIZE VARIABLE IN 2-D SHAPE SPACE.'

INPUT: It has no input. OUTPUT: PLOTS SHOWN IN FIG. 17.

```
DATA SIZE;
INPUT M1 M2 C D E F G H;
CARDS;
0       0       0       1       0         2        0     3
3       5       1       0       2         0       3      0
PROC GPLOT;
PLOT M1*M2=1 C*D=2 E*F=3 G*H=4/OVERLAY;
SYMBOL1 I=JOIN;
SYMBOL2 I=JOIN;
SYMBOL3 I=JOIN;
SYMBOL4 I=JOIN;
TITLE1 PLOT OF A TWO DIMENSIONAL SHAPE LINE;
TITLE2 SHOWING THE SIZE VARIABLE OF THE FORM M1+M2=CONST;
PROC MATRIX;
FETCH SIZE;
THETA=0;
DELTA=6.280/100;
CC=J.(100,8,.);
DO I=1 TO 100;
CC(I,1)=3*THETA;
CC(I,2)=5*THETA;
CC(I,3)=COS(THETA);;
CC(I,4)=.5*SIN(THETA);
CC(I,6)=1*SIN(THETA);
CC(I,5)=2*COS(THETA);
CC(I,7)=3*COS(THETA);
CC(I,8)=1.5*SIN(THETA);
THETA=THETA+DELTA;
END;
OUTPUT CC OUT=CCC (RENAME=(COL1=M1 COL2=M2 COL3=Z COL4=A
COL5=B COL6=C COL7=D COL8=E));
DATA NCC AND ;SET CCC;IF ((E >= 0) AND ( D >= 0));
PROC GPLOT DATA=NCC;
PLOT M1*M2=1 Z*A=2 B*C=3 D*E=4/OVERLAY;
SYMBOL1 I=JOIN;
SYMBOL2 I=SPLINE;
SYMBOL3 I=SPLINE;
SYMBOL4 I=SPLINE;
TITLE1 PLOT OF A TWO DIMENSIONAL SHAPE LINE;
TITLE2 USING THE SIZE VARIABLE OF THE FORM SQRT(M1 +A*M2 ) =CONSTANT;
```

FAST FOURIER TRANSFORM

and

RESAMPLING USING LINEAR INTERPOLATION.

INPUT: TO THE RESAMPLING USING LINEAR INTERPOLATION

A data set named SHAPEXY consisting of the x and y coordinates of the points between which the linear interpolation is to be carried out before resampling.

OUTPUT: OF THE RESAMPLING USING LINEAR INTERPOLATION:

A data set named SHAPE1 which has three columns. The first column is the sequence number of the sample while the next two are its x and y coordinates respectively. The program is designed to include the input data in the resamples. RES is a temporary array in the program which stores the resampled values. Depending upon the resampling interval ISAMP the dimensions of RES will have to be altered for every run.

INPUT: TO THE FAST FOURIER TRANSFORM PROGRAM:

A data set of the form similar to SHAPE1 described above.

OUTPUT: OF THE FAST FOURIER TRANSFORM PROGRAM: Plots of the real and imaginary parts of the transform, normalized by the factor s and σ respectively.

```
PROC MATRIX;
FETCH SHAPEXY;
NN=NROW(SHAPEXY);
MM=NN+1;
CRN=J.(MM,2,0);
CRN(1:NN,)=SHAPEXY;
CRN(MM,)=SHAPEXY(1,);
DIS=J.(NN,1,0);
RES=J.(300,2,0);


*CALCULATE THE LINEAR LENGTH;


DIJ=0;
 DO I=1 TO NN BY 1;
J=I+1;
A=CRN(I,1)-CRN(J,1);
B=CRN(I,2)-CRN(J,2);
C=A#A+B#B;
C=SQRT(C);
DIJ=DIJ+C;
DIS(I,)=C;
END;


*FIND THE RESAMPLING INTERVAL;


ISAMP=DIJ0/128;


*FIND THE POINTS IN BETWEEN;


KK=0;


DO I=1 TO NN BY 1;
KK=KK+1;
J=I+1;
ACL=CRN(J,1)-CRN(I,1);
ACL=ABS(ACL);
C1=CRN(J,1);
```

```
C2=CRN(J,2);
X0=CRN(I,1);
Y0=CRN(I,2);
RES(KK,)=CRN(I,);
NINT=DIS(I,)*/ISAMP;
NP=INT(NINT);
IF (ACL <-.0001)
THEN
DO K=1 TO NP BY 1;
KK=1+KK;
IF (C2 > Y0) THEN
YYY=Y0+ISAMP;
IF (C2 < Y0) THEN YYY=Y0-ISAMP;
RES(KK,1)=X0;
RES(KK,2)=YYY;
Y0=YYY;
GO TO LABEL1;
END;
ELSE
DO;
M=(CRN(J,2)-CRN(I,2))*/(CRN(J,1)-CRN(I,1));
C=CRN(I,2)-M*CRN(I,1);
DO K=1 TO NP BY 1;
KK=KK+1;
DSQ=ISAMP*ISAMP;
SSQ=(M*M)+1;
SPA=DSQ*/SSQ;
SPA=SQRT(SPA);
IF (X0 < C1) THEN
XXX=SPA+X0;
IF (X0 > C1) THEN XXX=X0-SPA;
YYY=M*XXX+C;
X0=XXX;
Y0=YYY;
RES(KK,1)=XXX;
RES(KK,2)=YYY;
END;
END;
LABEL1:
END;


AY=J.(KK,1,10);
RES(1:KK,1)=RES(1:KK,1)+AY;
AYY=J.(KK,1,1000);
RES(1:KK,2)=RES(1:KK,2)+AYY;


RES=RES*1;
```

```
OUTPUT RES OUT=SHAPE(RENAME=(COL1=X COL2=Y));
DATA SHAPE1;J=_N_;SET SHAPE;IF (X NE 0  & Y NE 0);
PROC PLOT DATA=SHAPE1;
PLOT Y*X;
TITLE PLOT OF THE SWEPT WING PLANE;
TITLE2 OF SIZE=B;



PROC MATRIX;
FETCH SHAPE1;
NN=NROW(SHAPE1);
TFF=J.(NN,2,0);
NX=1+(NN#/2);
NF=FLOOR(NX);
NB=NN-NF;
FEX=J.(NN,2,0);
FEY=J.(NN,2,0);
REX=SHAPE1(,2);
IMY=SHAPE1(,3);
FEX(1:NF,)=FFT(REX);
FEY(1:NF,)=FFT(IMY);
*;
*OPERATION FOLD;
*;
DO I=1 TO NB BY 1;
J=NN-I+1;
FEX(J,)=FEX(I,);
FEY(J,)=FEY(I,);
END;
*;
* OPERATION FLIP;
DO I=1 TO NB BY 1;
J=NN-I+1;
FEX(J,2)=-FEX(J,2);
FEY(J,2)=-FEY(J,2);
END;
*;
* OPERATION ADD COMPONENTS;
TFF(,1)=FEX(,1)-FEY(,2);
TFF(,2)=FEX(,2)+FEY(,1);
A0=J.(1,2,0);
EXPTR=J.(2,2,0);
FEX=TFF;
FEY=TFF;
MX=TFF(1,1);
MT=TFF(1,1);
INS=NN-1;
FEX(1,)=A0;
FEY(1,)=A0;
ITISODD=0;
ITISEVEN=0;
NS=NN#/2;
```

```
NS=NS-FLOOR(NS);
AR=TFF(2,1);
AI=TFF(2,2);
IF (NS NE 0)
THEN
DO;
FEY(NN,)=A0;
ITISODD=1;
BR=TFF(INS,1);
BI=TFF(INS,2);
END;
IF (NS EQ 0)
THEN
DO;
ITISEVEN=1;
BR=TFF(INS,1);
BI=TFF(INS,2);
END;
QAA=AR#/AI;
IF ( AR >= 0) THEN QAA=ATAN(QAA);
IF (AR < 0 AND AI >= 0) THEN QAA=3.1415926535-ATAN(QAA);
IF (AR < 0 AND AI < 0) THEN QAA=3.1415926535+ATAN(QAA);
QBB=BR#/BI;
IF (BR >= 0)THEN QBB=ATAN(QBB);
IF (BR < 0 & BI >=0) THEN QBB=3.1415926535-ATAN(QBB);
IF (BR < 0 & BI <0) THEN QBB=3.1415926535+ATAN(QBB);
PHI=(QAA+QBB)#/2;
ALPHA=(QAA-QBB)#/2;
PRINT PHI ALPHA;
S=AI#AI+BI#BI;
S=SQRT(S);
FSIZE=1#/S;
*;
*;
*;
*;
*ENTER MULTIPLICATION BY EXP(PHI + N ALPHA);
*;
*;
DO I=1 TO NN BY 1;
IF (I =< NF) THEN J=I-1;
IF (I > NF) THEN DO;
IF (ITISODD = 1) THEN J=I-NN;
IF (ITISEVEN =1) THEN J=I-NN-1;
                  END;
NALPHA =J#ALPHA;
PHINALPH=PHI+NALPHA;
REX=COS(PHINALPH);
IEX=SIN(PHINALPH);
EXPTR(1,1)=REX;
EXPTR(1,2)=IEX;
EXPTR(2,1)=-IEX;
EXPTR(2,2)=REX;
APP=FEX(I,);
```

```
FEX(I,)=APP*EXPTR;
END;
FEX=FEX#PSIZE;
*;
*;
*:
*;
*ENTER STANDARD DEVIATION;
*;
*;
SAVI=FEY#FEY;
MSAV=SAVI(,+);
STSIZE=MSAV(+,);
SSIZE=SQRT(STSIZE);
PRINT PSIZE S SSIZE;
AZQ=SHAPE1(,2:3);
MEAN=AZQ(+,)#/NN;
AZQ(,1)=AZQ(,1)-J.(NN,1)#MEAN(,1);
AZQ(,2)=AZQ(,2)-J.(NN,1)#MEAN(,2);
SS=AZQ#AZQ;
SSA=SS(,+);
SIG=SSA(+,);
SCA=NN#/SIG;
SCA=SQRT(SCA);
STD=1#/SCA;
PRINT SCA STD ;
FEY=FEY#/STD;
FREE REX IMY;
OUTPUT FEX OUT=FEX(RENAME=(COL1=RR COL2=IR));
OUTPUT FEY OUT=FEY(RENAME=(COL1=RI COL2=II));
OUTPUT TFF OUT=TFF(RENAME=(COL1=RRR COL2=III));
DATA NFEX;J=_N_;SET FEX;
DATA NFEY;J=_N_;SET FEY;
DATA NTFF;J=_N_;SET TFF;
PROC PRINT DATA=NFEX;
PROC PRINT DATA=NFEY;
PROC PRINT DATA=NTFF;
PROC PLOT DATA=NFEX;
PLOT RR*J;
TITLE PLOT OF THE REAL PART OF THE TRANSFORM OF THE REAL PART;
*;
PROC PLOT DATA=NFEX;
PLOT IR*J;
TITLE PLOT OF THE IMAGINARY PART OF THE REAL PART  OF THE
TRANSFORM;
*;
PROC PLOT DATA=NFEY;
PLOT RI*J;
TITLE PLOT OF THE REAL PART OF THE TRANSFORM OF THE IMAGINARY PART;
*;
PROC PLOT DATA=NFEY;
PLOT II*J;
TITLE PLOT OF THE IMAGINARY PART OF THE TRANSFORM OF THE IMAGINARY PART;
*;
```

```
PROC PLOT DATA=NTFF;
PLOT RRR*J;
TITLE PLOT OF THE REAL PART OF THE COMPLEX TRANSFORM;
*;
PROC PLOT DATA=NTFF;
PLOT III*J;
TITLE PLOT OF THE IMAGINARY PART OF THE COMPLEX TRANSFORM;
```

LISTING FOR THE PROGRAM USED TO ROTATE AND SHIFT THE DATA

INPUT: Two column input representating the data to be rotated.

OUTPUT: Two column output representating the rotated data

The input and output data set are both called SHAPEXZ.

```
PROC MATRIX;FETCH SHAPEX2;N=NROW(SHAPEX2);
A=SIN(.785);B=COS(.785);
C=SHAPEX2(,1);D=SHAPEX2(,2);
SHAPEX2(,1)=C#B-D#A+J.(N,1,1);
SHAPEX2(,2)=C#A+D#B;
OUTPUT SHAPEX2 OUT=SHAPEX2(RENAME=(COL1=X COL2=Y));
PROC PRINT DATA=SHAPEX2;
```

LISTING FOR THE PROGRAM USED TO CHANGE THE DIRECTION OF SCAN FROM CLOCKWISE TO ANTICLOCKWISE.

INPUT: Three column input data set called SHAPEXY. The first column is the sequence number while the last two columns represents the x and y coordinates of the data points.

OUTPUT: Data set SHAPEXZ with the stack of the input x and y coordinates inverted.

```
PROC MATRIX;
FETCH SHAPEXY;
N=NROW(SHAPEXY);
SET=1;
RESET=0;
IXI=N+1;
CLOCKWIS=RESET;
TEMPX=J.(N,3,.);
IF (CLOCKWIS=SET) THEN GO TO XXX;
DO I=1 TO N;
JXJ=IXI-I;
TEMPX(I,1)=SHAPEXY(I,1);
TEMPX(JXJ,2:3)=SHAPEXY(I,2:3);
END;
SHAPEXY=TEMPX;
XXX:
FREE TEMPX;
OUTPUT SHAPEXY OUT=SHAPEXY (RENAME=(COL1=J COL2=X COL3=Y));
```

THE LISTING OF THE PROGRAM FOR FINDING THE CURVATURE:

INPUT: The input which follows after the cards statement consists of two columns representating the x and y coordinates of the data points.

OUTPUT: The output is a data set named SXY consisting of eight columns. The first column represents the sequence number of the data points. The 2nd and the 3rd represent the input data. The fourth represents the magnitude of curvature. The 5th and the 6th represent the x and y components of the Vector Curvature. The 7th and the 8th column represent the location of the critical points which are defined as the points where the curvature is two times the mean curvature.

```
DATA AXY;
J=_N_;
IB=J-3;
INPUT X Y;
DX=DIF(X);
DY=DIF(Y);
MDXY=SQRT(DX*DX+DY*DY);
UTX=DX/MDXY;
UTY=DY/MDXY;
DUTX=DIF(UTX);
DUTY=DIF(UTY);
MUTY=SQRT(DUTX*DUTX+DUTY*DUTY);
IF MUTY=0 THEN DO;
NNX=0;
NNY=0;
END;
ELSE
DO;
NNX=DUTX/MUTY;
NNY=DUTY/MUTY;
END;
DNNX=LAG(NNX);
DNNY=LAG(NNY);
IF (((DNNX=0)&(DNNY=0))) THEN DO;
IF (((NNX=0)&(NNY=0)))  THEN DO;
PRJ=1;END;
ELSE DO;
PRJ=-1;END;
END;
ELSE DO;
IF (((NNX=0)&(NNY=0))) THEN DO;
PRJ=1;END;
ELSE DO;
S=NNX*DNNX+NNY*DNNY;
IF ((S=<0)) THEN DO;
PRJ=-1;END;
ELSE DO;
PRJ=1;END;
END;
END;
CARDS;
```

```
PROC MATRIX;
FETCH AXY;
N=NROW(AXY);
NN=N-3;
SXY=J(NN,8,0);
SXY(,1)=AXY(4:N,2);
SXY(,2)=AXY(4:N,3);
SXY(,3)=AXY(4:N,4);
SXY(,4)=AXY(4:N,11);
MEANN=SXY(+,4)#/NN;
SXY(,5)=AXY(4:N,9);
SXY(,6)=AXY(4:N,10);
SXY(,7)=((AXY(4:N,11)>=(2#MEANN));
SXY(,8)=SXY(,7);
SXY( ,7:8)=SIGN(SXY(,7:8));
SXY(,7:8)=SXY(,7:8)#AXY(4:N,3:4);
FREE AXY;
OUTPUT SXY OUT=SXY(RENAME=(COL1=SAM COL2=X COL3=Y COL4=MAGCURV COL5=CURV
X
 COL6=CURVY COL7=CRITX COL8=CRITY));
DATA CRITICAL;SET SXY;
PROC PRINT DATA=CRITICAL;

DATA CRITICA;SET SXY;DROP SAM X Y MAGCURV CURVX CURVY;
IF CRITX<>0;
PROC PRINT DATA=CRITICA;
```

LISTING OF THE PROGRAM NEEDED TO INTERFACE

THE ADAPTIVE LINE OF SIGHT OUTPUT

TO

THE POST PROCESSING STEP.

```
DATA INF1;SET INF;RENAME II=I;DROP XII YII J JJ XJJ YJJ LOC XLOC YLOC;
DATA INF2;SET INF;RENAME JJ=I;DROP XJJ YJJ J II XII YII LOC XLOC YLOC;
DATA INF3;SET INF;RENAME LOC=I;DROP XLOC YLOC J II XII YII JJ XJJ YJJ;
DATA YYAINF; SET INF1 INF2 INF3;
PROC SORT DATA=YYAINF; BY I;
PROC MATRIX;FETCH YYAINF;MITE=NROW(YYAINF);
MITEE=MITE-1;
HUUU=0;
DO KITE=1 TO MITE;
MII=KITE+1;MIII=KITE+2;
THERE:
NODO=KITE+HUUU;
IF (NODO=MITE) THEN GO TO MEOUT;
ATAY=YYAINF(KITE,1);BTAY=YYAINF(MII,1);
IF (ATAY=BTAY) THEN
DO;
YYAINF(MII:MITEE,)=YYAINF(MIII:M
*PROC PRINT DATA=AINF;
```

LISTING OF THE PROGRAM TO INTERFACE

THE ADAPTIVE LINE OF SIGHT OUTPUT

TO

THE PLOT ROUTINE.

INPUT: Data from the Adaptive Line of Sight program.

OUTPUT: PLOTS for shownning the critical points obtained using the Adaptive Line of Sight method.

```
*;
*;
*;
*;
DATA SHAPEX;J1=_N_;SET SHAPEXY;DROP J;
DATA INFO;J1=_N_;SET INF;IF (II NE 0);
DATA SHAPE;
MERGE SHAPEX INFO; BY J1;
PROC GPLOT DATA=SHAPE;
*PLOT X*Y=1 XII*YII=2 XLOC*YLOC=3 /OVERLAY HZERO VZERO;
PLOT Y*X=1 YII*XII=2 YLOC*XLOC=3/OVERLAY HZERO VZERO;
SYMBOL1 I=JOIN;
SYMBOL2 V=DIAMOND;
SYMBOL3 V=STAR;
TITLE1 PLOT OF THE PARTIAL PIG OF SIZE = 10XB;
TITLE2 CRITICAL POINTS OBTAINED USING THE ADAPTIVE LINE OF SIGHT METHOD;
FOOTNOTE1 TYPICAL MINIMAL SET;
FOOTNOTE2 NOTE THE PIG WAS SCALED DISPLACED AND ROTATED;
```

END

FILMED

11-85

DTIC